AD NUMBER
AD859289
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; APR 1969. Other requests shall be referred to Naval Air Systems Command, ATTN: AIR-6022, Washington, DC 20360.
AUTHORITY
USNASC ltr dtd 26 Oct 1971

AD 859289

Report No. MSD-P69-144

Contract No. 00019-68-C-02~~7~~ 47

COLLOCATION FLUTTER ANALYSIS STUDY

This document is subject to special export controls and transmittal to foreign governments or foreign nationals may be made only with prior approval of the Naval Air Systems Command (AIR[redacted] 602).

VOLUME IV.

COFA - COMPUTER PROGRAM TO PERFORM FLUTTER ANALYSIS BY THE COLLOCATION METHOD

APRIL 1969

MISSILE SYSTEMS DIVISION

HUGHES

HUGHES AIRCRAFT COMPANY

COFA

COLLOCATION FLUTTER ANALYSIS STUDY

VOLUME IV

COFA - COMPUTER PROGRAM TO
PERFORM FLUTTER ANALYSIS BY THE COLLOCATION METHOD

Prepared by Dynamics & Environments Section Personnel
Hughes Aircraft Company, Missile Systems Division
Contract No. 00019-68-C-0247

APRIL 1969

TABLE OF CONTENTS

ABSTRACT

A collocation solution of the flutter and vibration problems for a multiple component system is presented. The formulation utilizes structural, aerodynamic, and inertial characteristics in the form of matrices of structural and aerodynamic influence coefficients and a mass matrix, respectively, for each component. The use of a rigid-body modal matrix permits a general analysis for a system free in space with up to six rigid-body degrees of freedom.

The computer program provides the flutter or vibration solution for a system composed of as many as 20 flexible components with a maximum total of 49 collocation control points. An option is provided to vary the density as well as the reduced velocity. Another option is provided to yield the modes from a vibration analysis in a punched-card format for use in flutter analysis by modal methods.

SECTION 1

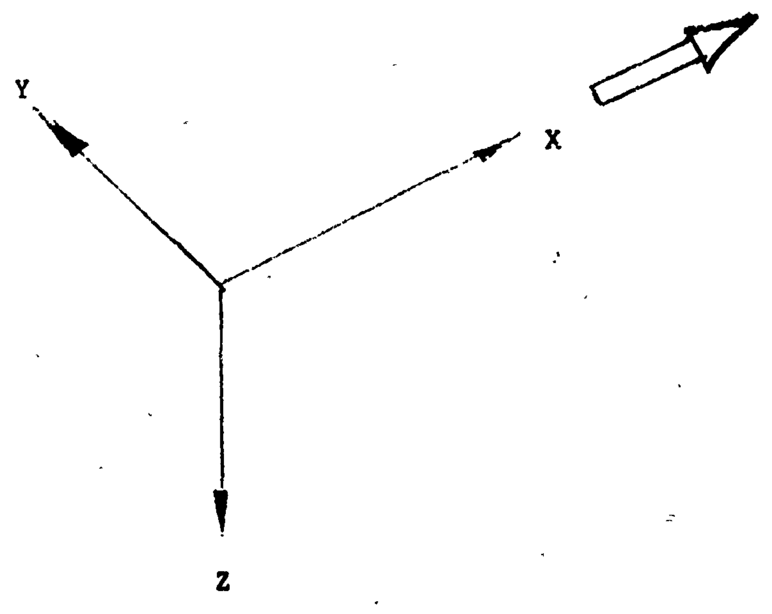
INTRODUCTION

The mathematical formulation of the flutter problem results in a set of integral equations whose closed form solution is impossible to obtain for most practical problems. One of the most useful approximate methods of solving these equations is by direct collocation. A solution by collocation is one in which the equations are satisfied at a finite number of selected points on the structure. These points, known as collocation points, are satisfied simultaneously. The collocation solution results in a matrix formulation which when cast in the canonical form will yield eigenvalues that are directly related to the flutter stability parameters. This manual presents a digital computer program that will perform collocation flutter analyses. The computer program, which is written in Fortran IV, was developed by Rodden, Farkas, and Malcom in Reference 1.

The collocation formulation of the flutter problem has been presented in Reference 2. The equations are presented for analysis of single component systems restrained (cantilevered) in space and for sysmmetrical systems free in space undergoing either symmetric or antisymmetric flutter. A method of generalizing the matrix equation for free-free flutter to include up to six rigid-body degrees of freedom has been given in Reference 3. The present program extends the formulation of Reference 2 to include an arbitrary combination of rigid-body degrees of freedom (Ref. 3), and to consider more than one flexible component. In addition, an option has been provided to vary the altitude (i.e. density) as well as the reduced velocity. Finally, options have been added to carry out a vibration analysis (which requires no aerodynamic data) and to provide vibration modes in punched-card format for use in a modal flutter or vibration analysis.

SECTION 2

The NASA body axis system with the x, y, and z axes directed forward, starboard, and downward, respectively, is recommended for consistency with the formulation of the static aeroelastic problems in Reference 4. However, the usual flutter convention with the x, y and z axes directed aft, starboard and downward, respectively, may be used instead. In either case, the vertical normal force and deflection are positive downward.



SECTION 3

DERIVATION OF EQUATIONS

The integral equations of aeroelasticity consist of two basic relationships: The first is the relation between the structural deformation, the structural influence function, and the inertial and aerodynamic loadings; the second is the relation between the aerodynamic disturbance (downwash), the aerodynamic influence (kernel) function, and the aerodynamic pressure. A collocation formulation of the deformation integral equation for a vehicle free in space may be written in matrix form by requiring that the integral equation be satisfied at a discrete set of control points. We choose a single type of coordinate, viz., the deflection h, as an adequate measure of both the deformation and the free-stream disturbance, not only for simplicity in the resulting equations but also because deflections have a more general meaning on a cambered vehicle and deflection influence coefficients are more readily obtained from a structural analysis than slope (or twist) influence coefficients. The resulting deformation matrix equation is

$$\{h_1\} - \{h_o\} = K[a]\,(\{F_i\} + \{F_a\}) \qquad (1)$$

where $\{h_1\}$ is the set of components of the absolute deflections of the control points, $\{h_o\}$ is the set of components of the deflections of the control points due to the rigid-body motion of some reference points, $[a]$ is the set of structural influence coefficients (SICs, or flexibility matrix) for the system <u>cantilevered</u> from (or otherwise restrained at) the reference point, $\{F_i\}$ is the set of inertial force components integrated throughout the region adjacent to each control point, $\{F_a\}$ is the set of aerodynamic force components integrated over the vehicle surface adjacent to each control point, and the scaler K has been introduced as a factor to the SICs for convenience in investigating variations in stiffness levels. The inertial forces may be written in terms of a mass matrix $[M]$ and the control point accelerations.

$$\{F_i\} = -(1/386)[M]\{\ddot{h}_1\} \tag{2}$$

where the diagonal elements of the mass matrix are found from integrating the structural mass density throughout the region adjacent to the control points. (N. B., the mass matrix need not be diagonal, and, in general, will not be so if the elements must be derived from a set of weight data previously lumped at a system of control points different from those required in the aeroelastic analysis. The use of a coupled mass matrix permits simulation of given inertial characteristics at a set of control points frequently dictated by more difficult aerodynamic considerations.)

A collocation formulation of the aerodynamic integral equation is more difficult than in the case of the deformation integral equation because of the singularities in the aerodynamic kernel function. The determination of three relationships is necessary to derive a set of aerodynamic influence coefficients (AICs) that relate the control point forces to the deflections. The most basic and difficult is the pressure-downwash relation that is derived from numerical analysis of the aerodynamic integral equation. The simpler relations are the numerical integration of the pressure to obtain the force, and the numerical substantial differentiation of the deflection to obtain the downwash. The effort involved in each step depends on the planform, Mach number regime, and frequency range; a survey of the status of unsteady AICs is given in Ref. 4. For present purposes, it is sufficient to state the definition of the AICs in the oscillatory case. We write the aerodynamic control point forces in terms of the control point deflections as

$$\{F_a\} = (4\pi^2/12)\rho f^2 b_r^2 s[W][C_h]\{h_1\} \tag{3}$$

where $[C_h]$ is the theoretically derived dimensionless (complex) matrix of oscillatory AICs, f is the frequency of the assumed harmonic motion, ρ is the atmospheric density, b_r is the reference semichord, s is the reference span, and $[W]$ is an empirically derived weighting matrix

for modification of the theoretical AICs. A method for obtaining the elements of the weighting matrix has been suggested in Ref. 4.

The sum of the force components may be written now from Eqs. (2) and (3) for the case of harmonic motion.

$$\{F_i\} + \{F_a\} = (4\pi^2 f^2/386)([M] + 32.174\ \rho b_r^2 s[W][C_h])\{h_1\} \quad (4a)$$

$$= (4\pi^2 f^2/386)[\overline{M}]\{h_1\} \quad (4b)$$

We next discuss the manner of inclusion of the rigid-body degrees of freedom in Eq. (1). The matrix $\{h_o\}$ has been defined as the set of components of the deflections of the control points due to the rigid-body motion of the reference point. Each component of the control point deflections h_o is linearly related to the rigid-body translations and rotations, provided the rotations are small. Therefore, we may define a rigid-body modal matrix $[h_R]$ as the transformation

$$\{h_o\} = [h_R]\{a_R\} \quad (5)$$

where $\{a_R\}$ is the set of amplitudes of rigid-body translations and rotations of the reference point. As an example, if we consider symmetrical vertical motion, $[h_R]$ is composed of two columns: the first is a unit column corresponding to the plunging mode, the second consists of the x-coordinate of each control point corresponding to the pitching mode; $\{a_R\}$ is composed of two elements: the first is the plunging displacement z_o, the second is the pitching angular displacement θ.

Before proceeding in the derivation, we should review the format of the various matrices in the case of a multiple flexible component system. As an example, we consider a symmetrical flutter analysis of an aircraft whose wing, aft fuselage, and tail are flexible, and whose forward fuselage may be assumed to be rigid. We assume that the reference point (cantilever point) can be located in the vehicle such that

its various components are independent. If we choose a point at the intersection of the wing and fuselage, then the wing is independent of the aft fuselage-tail combination, but the tail and aft fuselage must be considered together. The motion of the rigid forward fuselage is determined by the motion of the reference point, and the forward fuselage will not enter into any flexible considerations but only into the free-free boundary conditions. From the foregoing, it is seen that the various matrices will appear in partitioned form. If we denote the wing and aft fuselage-tail system by the subscripts 1 and 2, respectively, then the flexibility matrix appears as

$$[a] = \left[\begin{array}{c|c} a_1 & 0 \\ \hline 0 & a_2 \end{array}\right] \tag{6}$$

the mass matrix as

$$[M] = \left[\begin{array}{c|c} M_1 & 0 \\ \hline 0 & M_2 \end{array}\right] \tag{7}$$

the weighting matrix as

$$[W] = \left[\begin{array}{c|c} W_1 & 0 \\ \hline 0 & W_2 \end{array}\right] \tag{8}$$

the AICs as

$$[C_h] = \left[\begin{array}{c|c} C_{h1} & 0 \\ \hline 0 & C_{h2} \end{array}\right] \tag{9}$$

and the rigid-body modal matrix as

$$[h_R] = \left[\begin{array}{c} h_{R1} \\ \hline h_{R2} \end{array}\right] \tag{10}$$

Two requirements should be emphasized with regard to the AICs. The first concerns the proper inclusion of the reference geometry associated with the nondimensional AICs. The dimensional form of Eq. (9) may be written

$$b_r^2 s[C_h] = \left[\begin{array}{c|c} b_1^2 s_1 C_{h1} & 0 \\ \hline 0 & b_2^2 s_2 C_{h2} \end{array}\right] \tag{11}$$

where b_r and s are the reference semichord and span of the composite system, b_1 and s_1 are the reference geometry for the first component, and b_2 and s_2 are the reference geometry for the second component. The second requirement is that the AICs for each component must be determined for the same "dimensional" reduced velocity V/ω. If the reference reduced velocity is

$$1/k_r = V/b_r\omega \tag{12}$$

then the reduced velocity for the first component must be

$$1/k_1 = (1/k_r)(b_r/b_1) \quad (13)$$

and, for the second component,

$$1/k_2 = (1/k_r)(b_r/b_2) \quad (14)$$

Both of these requirements can be met in formulating the composite AICs by choosing the same reference geometry in determining the AICs for each component.

The rigid-body modal matrix provides the basis for a general statement of the boundary conditions for the free-free flutter of the composite system. The boundary conditions for harmonic motion may be written as

$$[h_R]^T [\overline{M}] \{h_1\} + [\Delta\overline{m}] \{a_R\} = \{0\} \quad (15)$$

where $[\Delta\overline{m}]$ is an incremental generalized mass matrix, including aerodynamic effects, of any rigid component of the system attached to the reference point (e. g., the forward fuselage that was assumed to be rigid in the foregoing example),* and is not considered in the formulation of the flexible component mass and aerodynamic matrices. The form of the matrix $[\Delta\overline{m}]$ may be illustrated by the previous example with a rigid forward fuselage again in symmetrical motion. We may wri

$$[\Delta\overline{m}] = [\Delta m] + [\Delta Q] \quad (1\cdot\cdot)$$

*N. B.: It is assumed that no dynamic coupling exists between the rigid and flexible components. A suitable distinction can always be made between the rigid and flexible components such that this requirement can be met.

NOTICE TO USERS

Portions of this document have been judged by the Clearinghouse to be of poor reproduction quality and not fully legible. However, in an effort to make as much information as possible available to the public, the Clearinghouse sells this document with the understanding that if the user is not satisfied, the document may be returned for refund.

If you return this document, please include this notice together with the IBM order card (label) to:

Clearinghouse
Attn: 152.12
Springfield, Va. 22151

where the generalized rigid component mass matrix of the forward fuselage is

$$[\Delta m] = \begin{bmatrix} M_o & S_o \\ S_o & I_{yo} \end{bmatrix} \tag{17}$$

in which M_o, S_o, and I_{yo} are the mass, static unbalance about the reference point, and pitching moment of inertia about the reference point, respectively, of the forward fuselage, and the generalized aerodynamic forces on the forward fuselage (if not negligible) are found from

$$[\Delta Q] = 32.174\ \rho b_o^2 s_o [h_{Ro}]^T [C_{ho}] [h_{Ro}] \tag{18}$$

where $[h_{Ro}]$ is the rigid-body modal matrix, $[C_{ho}]$ is the set of AICs, and b_o and s_o are the reference geometry for the forward fuselage. Again the AICs must be found for the reduced velocity of the composite system.

We are now in a position to eliminate the rigid-body degrees of freedom and to formulate the eigenvalue problem for the flutter of the flexible free-free system. Substituting Eqs. (4b) and (5) into Eq. (1), and adding the structural damping factor $1/(1 + ig)$ to the flexibility matrix to provide the artificial structural damping necessary to sustain the assumed harmonic motion of the flutter system, we obtain

$$\{h_1\} - [h_R]\{a_R\} = (4\pi^2 Kf^2/386(1 + ig))[a][\overline{M}]\{h_1\} \tag{19}$$

or

$$\lambda(\{h_1\} - [h_R]\{a_R\}) = [a][\overline{M}]\{h_1\} \tag{20a}$$

$$= [U]\{h_1\} \tag{20b}$$

where λ denotes the eigenvalue

$$\lambda = \lambda_R + i\lambda_I \tag{21a}$$

$$= 386(1 + ig)/4\pi^2 K f^2 \tag{21b}$$

Premultiplying Eq. (20b) by $[h_R]^T [\overline{M}]$, and multiplying Eq. (15) by λ and subtracting, permits solution for the amplitudes of the rigid-body motion

$$\lambda\{a_R\} = -[\overline{m}]^{-1}[h_R]^T[\overline{M}]\,[U]\{h_1\} \tag{22}$$

where

$$[\overline{m}] = [h_R]^T[\overline{M}]\,[h_R] + [\Delta\overline{m}] \tag{23}$$

Finally, substituting Eq. (22) into Eq. (20b) yields the generalized matrix equation for free-free flutter

$$\lambda\{h_1\} = ([I] - [h_R]\,[\overline{m}]^{-1}[h_R]^T[\overline{M}])[U]\,\{h_1\} \tag{24}$$

The solution of Eq. (24) for the complex eigenvalues leads to the free-free frequency and the required structural damping. From Eq. (21), we obtain the frequency

$$f = (1/2\pi)\sqrt{386/K\lambda_R} \tag{25}$$

and the required structural damping

$$g = \lambda_I/\lambda_R \tag{26}$$

Since the formulation of the AICs requires the assumption of a reduced velocity $1/k_r$, the velocity follows from that and the frequency obtained in Eq. (25)

$$U = 0.5921\,(2\pi f b_r)(1/k_r) \tag{27}$$

Equation (24) is seen to be completely general, being applicable from the cantilever case (in which we let $[h_R]$ vanish) to the case of six rigid body degrees of freedom, and for a vibration analysis for which the aerodynamic terms and required structural damping are deleted. We observe that Eq. (24) is a matrix formulation of the algebraic procedures for free-free vibration analysis described in Ref. 4 (Par. 11.2).

From a series of solutions of Eq. (24) for various reduced velocities, the conventional required damping versus velocity stability curve can be constructed for a specific altitude, and the flutter point is determined as the velocity for which the required damping and actual damping are equal. An alternative approach to the flutter analysis is based on a single representative reduced velocity and a series of solutions of Eq. (24), carried out for various densities. The density at which the required damping and actual damping are equal may be used to find a stiffness-altitude similarity parameter for flutter from which the flutter stability may be determined. However, at present, the validity of this latter approach requires further investigation, particularly the sensitivity of the results to the choice of representative reduced velocity.

The generalized mass corresponding to each free vibration mode is of interest in various modal analyses of flying qualities, stability and control characteristics, or transient response of flexible vehicles. If Eq. (24) is solved for the free vibration modes (by deleting the aerodynamic terms) then the n'th generalized mass is given by

$$m^{(n)} = \{h_1^{(n)}\}^T[M]\{h_1^{(n)}\} + \{a_R^{(n)}\}^T[\Delta m]\{a_R^{(n)}\} \tag{28}$$

where $\{h_1^{(n)}\}$ is the n'th free vibration mode and the corresponding rigid component mode is found from Eq. (22)

$$\{a_R^{(n)}\} = -\frac{4\pi^2 K}{386} f_n^2 [M]^{-1} [h_1]^T [M][U]\{h_1^{(n)}\} \qquad (29)$$

SECTION 4

REFERENCES

1. W. P. Rodden, E. F. Farkas and H. A. Malcom. "Flutter and Vibration Analysis by a Collocation Method: Analytical Development and Computation Procedure". Aerospace Corporation Report No. TDR-169(3230-11)TN-14, 31 July 1963.
2. W. P. Rodden, "Matrix Approach to Flutter Analysis" Institute of the Aerospace Sciences Fairchild Fund Paper No. FF-23, May 1958; based on North American Aviation, Inc., Report NA-56-1070, 1 May 1956.
3. W. P. Rodden, "On Vibration and Flutter Analysis with Free-Free Boundary Conditions". Journal of the Aerospace Sciences, 28 (1961) 65.
4. W. P. Rodden and J. D. Revell. "The Status of Unsteady Aerodynamic Influence Coefficients". Institute of Aerospace Sciences Fairchild Fund Paper No. FF-33, 23 January 1962; preprinted in Aerospace Corporation Report TDR-930(2230-09) TN-2, 22 November 1961.
5. R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. Aeroelasticity. Reading: Addison-Wesley Publishing Company, Inc., 1955.
6. R. H. Scanlan and Robert Rosenbaum. Introduction to the Study of Aircraft Vibration and Flutter. New York: The MacMillan Company, 1951.

SECTION 5

DESCRIPTION OF PROGRAM INPUT

UNITS

The dimensional data required for each component consist of the mass matrix in pounds, the flexibility matrix in inches per pound, and the reference semichord and semispan in feet. The aerodynamic influence coefficients are dimensionless. In the case of free-free analysis, the rigid-body mass matrix, e.g., in a symmetrical analysis if S_o and I_{yo} are given in the foot-pound system, the x-coordinates which correspond to the rigid body pitching mode must be measured in feet, whereas if S_o and I_{yo} are given in the inch-pound system, the x-coordinate must be measured in inches. The density is required in slugs per cubic foot.

CLASSES OF DATA AND PROBLEMS

Five classes of data must be provided: mass, aerodynamic influence coefficients (AICs) and their weighting matrices, structural influence coefficients, the rigid-body motion modal matrix, and the rigid component generalized mass characteristics. The cantilever case does not require either the rigid body modal matrix or the generalized masses. (For a vibration analysis, the aerodynamic input is not required.)

Several classifications of problems may be analyzed using the collocation flutter analysis program. They are cantilever flutter analysis — the structure is restrained from plunging motion. Free-free flutter analysis the structure is free to pitch, plunge, and roll. The free-free cases may have rigid components and flexible components. However, when flexible components are coupled together, the structural attachment between component must be statically determinate. When rigid body components are used, any number up to six rigid body modes may be used. Also vibration analyses may be performed when zero aerodynamic forces are used.

PROGRAM RESTRICTIONS AND OPTIONS

1. The maximum number of control points that can be used for all flexible components of any system is 49. A maximum of 49 control points may also be used for the rigid component for the purpose of deriving the generalized aerodynamic force.

2. The maximum number of flexible components is 20.

3. The maximum number of values used in the reference reduced velocity ($1/K_R$) series is 20.

4. The maximum number of values used in a density series is 20.

5. The program provides for varying the densities with each ($1/K_R$) or for using the same densities with all ($1/K_R$)'s.

6. The maximum number of output modes is 25.

7. The maximum number of rigid-body motion modes is 6.

8. It is possible to reserve a partition in the upper left-hand corner of the flexible components AIC matrices for control points whose aerodynamic forces may be neglected or found from an alternate theory to that used for the primary control points. This partition is termed the external stores region since external stores are an example of a source of additional control points requiring such special consideration. The maximum number of control points that can be reserved on each flexible component for external stores is 48.

9. A weighting matrix is an optional input for each flexible component. The order of this matrix must be identical with the order of the AIC matrices for the particular component.

10. Any number of complete sets (decks) of input data may be stacked and run in one machine pass.

DATA DECK SETUP

Loading Order

The data decks are assembled using cards punched from keypunch forms and/or card that are punched-card output from appropriate computer programs. The data items in each deck have the following order,

with the exception that some of the items may be omitted if indicators used in the control cards specify their absence.

1. Title card
2. Data deck general control card
3. K card (flexibility matrix scale factor)
4. Data card for change in matrix iteration tests
5. Control card(s) for external stores and weighting matrices
6. Reference semichord (b_r) and reference reduced velocities $(1/k_r)$'s
7. Reference semichord (b_{ri}) and reference semispan (S_i) for rigid and flexible components (surfaces)
8. Density series (if same densities are used for all $(1/k_r)$'s
9. Generalized mass matrix ($[\Delta m]$) for rigid components
10. Mass matrix ($[M]_i$) for each flexible surface
11. Rigid-body motion modal matrix ($[h_{R_O}]$) for rigid component
12. Rigid-body motion modal matrix ($[h_{R_i}]$) for each flexible surface
13. Flexibility matrix ($[a]_i$)
14. Weighting matrices ($[W]_i$)
15. Aerodynamic input repeated for each $(1/k_r)$
    a. Density series cards (if densities vary with each $(1/k_r)$)
    b. AIC matrix ($[C_{h_O}]$) for rigid component (if present)
    c. AIC matrix ($[C_h]_i$) for each flexible surface

Input Data Description

1. The title card may contain any alphanumeric characters desired in Columns 2 through 72.

2. Data deck general control card (FORMAT 18I4):

Col	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-32	33-36
Name	NSUR	NAERO	NRIGID	NFUS	NDENS	MODES	NDELM	NPUNCH	NCOM
Field	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

NSUR: Number of flexible components (surfaces), ≤ 20.

NAERO: Number of reference reduced velocities, ≤ 20; NAERO = 0 is used for vibration analyses.

NRIGID: Number of rigid-body motion modes to be input (= number of columns in ($[h_R]$); NRIGID = 0 for the cantilever case.

NFUS: NFUS must = 1 if AICs ($[C_{ho}]$)$_j$, j = 1, NAERO are input for the rigid components; NFUS = 0 if $[C_{ho}]$ is not input.

NDENS: If NDENS = 0 the densities are to vary with each $1/k_r$ and are input as part of Item 15; if NDENS>0 this number of densities must be input as Item 8, and each density value will be used for all $(1/k_r)$'s.

MODES: Number of output modes, ≤ 25.

NDELM: NDELM = 0 if no rigid component generalized mass matrix ($[\Delta m]$)is input; NDELM = 1 if $[\Delta m]$ is input.

NPUNCH: This indicator is used to obtain a printout of the dynamic matrix ($[U]$) and to obtain the frequencies and modes in punched-card format; NPUNCH = 0 if no printout of $[U]$ or punched output is desired; NPUNCH = 1 if only punched-card output is desired and NPUNCH = -1 will provide the printout of $[U]$ and the punched output. (The minus sign must be in Column 29 and the 1 (one) in Column 32.)

NCON: This indicator provides for changing five program test numbers used in the matrix iteration subroutine; NCON = 0 if no changes are desired and NCON $\neq$ 0 if any of the tests is to be changed.

3. K card (FORMAT 6E12.8)

Col	1-2
Name	K
Field	(1)

Field 1 contains K, the flexibility matrix normalizing constant; if the [a] matrix has not been normalized, enter K = 1.0. The flexibility matrix calculated by the program FLUENC has not been normalized.

4. Data card for changes in matrix iteration tests (FORMAT 3E12.8 and 2I4). Omit this card when NCON = 0. There are three test numbers and two control numbers that define the convergence criteria for the flutter eigenvalue solution. A suggested set of numbers are built into the program; these, however, may be changed by the program user. To alter any number, all five numbers must be reentered.

Col	1-12	13-24	25-36	37-40	41-44	
Name	EPSP	EPDP	AITKEN	NITRSP	NITRDP	
Field	(1)	(2)	(3)	(4)	(5)	

EPSP = 0.5×10^{-6} or input number; test for eigenvector convergence when the iteration procedure is approaching a single root.

EPDP = 0.5×10^{-7} or input number; test for convergence when the interation procedure is approaching a pair of close roots.

AITKEN = 0.9 or input number; if this test is met, the program uses a procedure (known as Aitken's δ^2 method) to accelerate convergence.

NITRSP = 40; a maximum of 40 single precision arithmetic iterations will be performed for each eigenvalue if its eigenvectors have not converged in a lesser number.

. ITRDP = 100; if the eigenvectors for any one eigenvalue have not converged in NITRSP single precisions iterations, the program will then use up to a maximum of 100 double arithmetic iterations.

5. Control card(s) for external stores and weighting matrices (FORMAT 18I4); omit this data when NAERO = 0.

Col	1-4	5-8	9-12	13-16	
Name	$ISXT_1$	ISW_1	$ISXT_2$	ISW_2	
Field	1	2	3	4	

Continue on successive cards to $ISW_i = ISW_{NSUR}$.

$ISXT_i$ = Number of control points reserved for the external stores on surface i.

ISW_i = 0, no weighting matrix is to be input for surface i.
= 1, weighting matrix is to be input for surface i.

Continue on next card, until i = NSUR.

6. Reference semichord, b_r and reduced velocity $(1/k_r)$ series (FORMAT 6E12.8): These reference parameters are used in computing the flutter velocities.

Col	1-12	13-24	25-36	37-48	49-60	61-72
Name	b_r	$(1/k_r)_1$	$(1/k_r)_2$	$(1/k_r)_3$	$(1/k_r)_4$	$(1/k_r)_5$
Field	(1)	(2)	(3)	(4)	(5)	(6)

Continue $(1/k_r)_i$'s on next card(s); $i \leq 20$. The b_r(feet) may be any value; but it is noted that the $(1/k_r)_i$ predetermines the $(1/k_i)$ used when computing the AIC matrices for each surface. The $1/k_i$ for surface i is found from the relationship $(1/k_i) = (1/k_r)(b_r/b_{ri})$ where b_{ri} is the reference semichord for surface i.

7. A reference semichord (b_{r_i}) and semispan (s_i) must be input for the rigid component if NFUS = 1, and for each flexible surface if NAERO>0.

When NFUS = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	b_{r_0}	s_{r_0}	b_{r_1}	s_{r_1}	b_{r_2}	s_{r_2}
Field	(1)	(2)	(3)	(4)	(5)	(6)

Where b_{r_0} and s_{r_0} are the reference semichord and semispan for the rigid component. Continue on next card(s) until $b_{r_i} = b_{r_{NSUR}}$ and $s_{r_i} = s_{r_{NSUR}}$.

When NFUS = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	b_{r_1}	s_{r_1}	b_{r_2}	s_{r_2}	b_{r_3}	s_{r_3}
Field	(1)	(2)	(3)	(4)	(5)	(6)

Continue on next card(s) until $b_{r_i} = b_{r_{NSUR}}$ and $s_{r_i} = s_{r_{NSUR}}$.

8. Density series (FORMAT 6E12. 5): Omit this input if NDENS=0 in Item 2. If NDENS>0 begin in field 1 of this card and NDENS densities, ≤ 20. Continue on successive card(s).

Column	1-12	13-24	25-36	37-48
Name	ρ_1	ρ_2	ρ_3	ρ_4
Field	1	2	3	4

9. Rigid component generalized mass matrix $[\Delta m]$.The $[\Delta m]$ matrix for the rigid component(s) of the system must be compatible with the flexible surfaces product matrix given by $[h_R]^T[M][h_R]$, i.e., each element in $[\Delta m]$ is based upon the same rigid-body motion generalized coordinate as the respective element in the product matrix. Input by column beginning each column on a new card. Omit this data when NDELM = 0.

Col	1-12	13-24	25-36	...		
Name	$\Delta m_{1,1}$	$\Delta m_{2,1}$	$\Delta m_{3,1}$		$\Delta m_{(NRIGID-1),1}$	$\Delta m_{NRIGID,1}$
Field	(1)	(2)	(3)		(NRIGID-1)	(NRIGID)

Continue on successive card(s) until

10. Mass matrix [M]: The mass matrix is partitioned as shown on page A-10, only the nonzero partitions are input; i.e., a separate mass matrix ($[M_i]$) is input for each surface. The sequence for considering the surfaces is the same as that used in Item 5 and 7 if NAERO>0. Repeat the following input for each surface from i = 1 to NSUR.

Size Control Cards

Column	1-4	5-8	9-12	
Name	$NSIZE_i$			
Field	(1)	(2)	(3)	

$NSIZE_i$ = the order of $[M]_i$

Often many of the elements in a mass matrix are zero; the following format has been provided so that most of the zero elements will not need to be entered into the program as data.

Control Card(s) for Omitting Zeros (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	17-20	...	
Name	LOW_1	$LHIGH_1$	LOW_2	$LHIGH_2$	LOW_3		$LHIGH_i$
Field	(1)	(2)	(3)	(4)	(5)		(2_i)

LOW_i = The row number in which the first nonzero element appears in Column i.

$LHIGH_i$ = The row number in which the last nonzero element appears in Column i.

If only one nonzero element appears in Column i (i.e., a diagonal mass matrix) the row number in which it appears must be used for both LOW and LHIGH.

Mass Matrix Elements (FORMAT 6E12.8)

Col	1-12	13-24	25-36		
Name	$\Delta m_{1,LOW}$	$\Delta m_{1,LOW+1}$	$\Delta m_{1,LOW+2}$	$\Delta m_{1,LHIGH-1}$	$\Delta m_{1,LHIGH}$
Field	(1)	(2)	(3)		

The elements are entered by column; each column begins on a new card. Any zero elements in rows between LOW and LHIGH must be entered or their respective fields left blank. If external stores are present (ISXT>0) all store control points must be entered before the surface control points; i.e., the elements representing the external stores mass must occupy the upper left-hand corner of the mass matrix.

11. Rigid component modal matrix, $[h_{R_0}]$ (see page 7). Omit this input when NFUS = 0. The number of rows in $[h_{R_0}]$ must be the same as the number of control points considered when computing the $[C_{h_c}]$ matrices; the number of columns must agree with NRIGID.

Size Control Card(s) (FORMAT 18I4)

Column	1-4	5-8	
Name	NROWS		
Field	(1)	(2)	

NROWS = The number of rows in $[h_{R_0}]$

Matrix $[h_{R_o}]$ Elements (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	$h_{R_{o1,1}}$	$h_{R_{o2,1}}$	$h_{R_{o3,1}}$	....	....	$h_{R_{oNROWS,1}}$
Field	(1)	(2)	(3)	(4)	(5)	(6)

The elements are entered by column, with each column beginning on a new card.

12. Rigid-body modal matrix, $[h_R]$ (see page 7). Omit this input if NRIGID = 0. $[h_R]$ is to be input by partitions $[h_R]_i$, each partition is of order (NSIZE x NRIGID) for each surface. The following data is to be repeated for i = 1, NSUR.

Size Control Card (FORMAT 18I4)

Column	1-4	5-8	
Name	NSIZE		
Field	(1)	(2)	

NSIZE = Number of control points on each surface

Matrix $[h_R]_i$ Elements (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	49-60	61-72
Name	$h_{R_{1,1}}$	$h_{R_{2,1}}$	$h_{R_{3,1}}$	....	....	$h_{R_{NSIZE,1}}$
Field	(1)	(2)	(3)	(4)	(5)	(6)

The elements are entered by column, with each column beginning on a new card.

13. Flexibility matrices, [a] (see page 6). The flexibility matrix is partitioned, only the nonzero partitions $[a]_i$ corresponding to the

flexible surfaces are entered. The matrix may be formed by any of the well-known procedures using elementary beam theory, force or displacement methods. The program FLUENC will generate this matrix using the displacement method. The punched output from FLUENC may be used as direct input into this program. The following data is repeated for i = 1, NSUR.

Control Card (FORMAT 18I4):

Column	1-4	5-8	9-12	13-16
Name	m_i	(BLANK)	IFORM	IROW
Field	(1)	(2)	(3)	(4)

m_i = The number of rows in $[a]_i$

IFORM = 0 if the elements are to be input using column binary format.

= 1 if the elements are to be input using FORTRAN (FORMAT 6E12.8) or FLUENC output is to be used directly.

IROW = 0 if the matrix elements are to be entered by column.

= 1 if the matrix elements are to be entered by row.

Matrix $[a]_i$ elements (use format specified above):

For IFORM = 1 and IROW = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48
Name	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$
Field	(1)	(2)	(3)	(4)

Each row starts on a new card.

For IFORM = 1 and IROW = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	$a_{1,1}$	$a_{2,1}$	$a_{3,1}$	$a_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If IFORM = 0, then IROW must = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location $(1 + m_i)$; Column 3 starts in location $(1 + 2m_i)$; etc. A TRA* card must end each $[a_i]$ deck. (The column binary format should be used only if the data are available as punched-card output from appropriate computer programs.) The only advantage of the C-B format is the minimum card storage space required.

14. Weighting matrix, [W] (see page 6). The weighting matrix is partitioned, only the nonzero partitions $[W]_i$ corresponding to the flexible surfaces are entered. No provisions have been made for entering a $[W_0]_i$ matrix for the rigid component; any adjustment to $[C_{h_0}]$ must be made before it is input as data. If $ISW_i = 0$ omit this data. Repeat the following data for i = 1, NSUR.

For $(ISW)_i = 0$ and $(ISXT)_i > 0$

Control card for external stores elements (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	$NXST_i$	(BLANK)	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

*The TRA card has a 7 and 9 punch in Column 1, Column 2 through 72 are blank and Column 73 through 80 will contain the characters used for identification and sequencing in the punched card output deck.

$NXST_i$ = 0 if no $[W]_i$ matrix is input for the external stores area (the program will use a unit matrix, I)

= n the number of control points reserved for stores.

NFORM = 1 if the $[W]_i$ matrix elements will be input using FORTRAN (FORMAT 6E12.8)

= 0 if the elements are to be input using column binary format

NROW = 0 if the $[W_i]$ matrix elements are to be input by column

= 1 if the matrix elements are [illegible] input by row

External stores elements $[W]_i$. Format given on control card above.

For NFORM = 1 NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{1,2}$	$W_{1,3}$	$W_{1,4}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

For NFORM = 1 NROW = 0

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{2,1}$	$W_{3,1}$	$W_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If NFORM = 0, then must NROW = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location ($1 + IXST_i$); Column 3 starts in location ($1 + 2IXST_i$); etc. A TRA card must end each $[a_i]$ deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

Control card for flexible surface weighting matrix $[W]_i$

(FORMAT 18I4) The $[W_i]$ matrix is often sparse, sometimes diagonal and may be of large order, ≤49; for this reason we provide for partitioning of the matrix and entering only the nonzero partitions.

Column	1-4	5-8	9-12	13-16	
Name	$NSIZE_i$	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = The number of control points used on surface i. Do not include control points for external stores.

NPART = The number of partitions in the $[W_i]_j$ surface matrix

NFORM = 1 if the $[W]_i$ will be input using FORTRAN (FORMAT 6E12.8)

= 0 if the elements are to be input using column binary format

NROW = 0 if the $[W_i]$ matrix elements are to be input by column

= 1 if the matrix elements are to be input by row

Repeat the following two data item for each partition j = 1, NPART.

Control card for partition $[W_i]_j$ FORMAT (18I4)

Column	1-4	5-8	9-12	13-16	17-21	
Name	N_j					
Field	(1)	(2)	(3)	(4)	(5)	

N_j = The order of partition j of $[W_i]$

Elements in partition $[W_i]_j$. Format given on the control card for flexible surface weighting matrix.

NFORM = 1 NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,1}$	$W_{1,2}$	$W_{1,3}$	$W_{1,4}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

NFORM = 1 NROW = 0

Column	1-12	13-24	25-36	37-48	
Name	$W_{1,2}$	$W_{2,1}$	$W_{3,1}$	$W_{4,1}$	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card.

If NFORM = 0, then must NROW = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location (1 + $IXST_i$); Column 3 starts in location (1 + $2IXST_i$) etc. A TRA card must end each $[a_i]$ deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

For $(ISW)_i = 0$ and $(ISXT)_i = 0$

Control card for flexible surface weighting matrix $[W]_i$ (FORMAT 18I4). The $[W_i]$ matrix is often sparse, sometimes diagonal and may be of large order, ≤ 49; for this reason we provide for partitioning of the matrix and entering only the nonzero partitions. Repeat the following data for i = 1, NSUR.

Column	1-4	5-8	9-12	13-16	17-20	
Name	$NSIZE_i$	NPART	NFORM	NROW		
Field	(1)	(2)	(3)	(4)	(5)	

NSIZE = The number of control points used on surface i

NPART = The number of partitions in the $[W_i]_j$ surface matrix

NFORM = 1 if the $[W]_i$ will be input using FORTRAN (FORMAT 6E12.8)

= 0 if the elements are to be input using column binary format

NROW = 0 if the elements are to be input by column

= 1 if the elements are to be input by row

Repeat the following two data items for each partition j = 1, NPART.

Control card for partition $[W_i]_j$ FORMAT (18I4)

Column	1-4	5-8	9-12	13-16
Name	N_j			
Field	(1)	(2)	(3)	(4)

N_j = The order of partition j of $[W_i]$

Elements in partition $[W_i]_j$ format given on the control card for flexible surface weighting matrix.

NFORM = 1   NROW = 1

Column	1-12	13-24	25-36	37-48
Name	$W_{1,1}$	$W_{1,2}$	$W_{1,3}$	$W_{1,4}$
Field	(1)	(2)	(3)	(4)

Each row starts on a new card.

NFORM = 1   NROW = 0

Column	1-12	13-24	25-36	37-48
Name	$W_{1,1}$	$W_{2,1}$	$W_{3,1}$	$W_{4,1}$
Field	(1)	(2)	(3)	(4)

Each column starts on a new card.

If NFORM = 0, then must NROW = 0: The matrix elements are input using column binary format; Column 1 starts in Origin 1. Column 2 starts in location $(1 + m_i)$; Column 3 starts in location $(1 + 2m_i)$; etc. A TRA card must end each $[a_i]$ deck. The column binary format should

used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

15. Aerodynamic data: The aerodynamic input consists of NAERO sets of AIC's. Each set of AIC's consists of the AIC's for each surface which have the same reference $1/k_r$ (see Item 6). If NDENS = 0 a density series will precede each set of AIC's. Input the density series (if Item 8 was omitted) and the AIC matrices for each surface with the following input order. Repeat the order for each $(1/k_r)_j$ j = 1, NAERO.

Control card for density series (FORMAT 18I4). Omit this input if NDENS>0.

Column	1-4	5-8	9-12	13-16	
Name	NRHO				
Field	(1)	(2)	(3)	(4)	

NRHO = The number of densities to be entered for $(1/k_r)_j \leq 20$

Density Series (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	ρ_1	ρ_2	ρ_3	ρ_4	
Field	(1)	(2)	(3)	(4)	

Rigid component AIC matrix $[C_{h_o}]_j$. Omit this input if NFUS = 0. The $[C_{h_o}]_j$ matrix may be sparse; thus, provision has been made to partition the matrix and enter only the nonzero partitions.

Reference 1/k for $[C_{h_o}]_j$ (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	1/k				
Field	(1)	(2)	(3)	(4)	

1/k = The reduced velocity used in computing the rigid component $[C_{h_o}]_j$. The AIC's in $[C_{h_o}]$ must be computed for a 1/k which properly relates them to the j^{th} $1/k_r$.

Control card for $[C_{h_o}]_j$ matrix (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	NSIZE	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = Number of control points on the rigid component

NPART = Number of nonzero partitions in $[C_{h_o}]_j$
= 1 for an unpartitioned matrix

NFORM = 1 if the $[C_{h_o}]_j$ matrix is to be input using FORTRAN (FORMAT 6E12.8)
= 0 if the $[C_{h_o}]_j$ matrix is to be input using column binary format

NROW = 1 if the $[C_{h_o}]_j$ matrix is to be input by row
= 0 if the $[C_{h_o}]_j$ matrix is to be input by column

Repeat the following data for each partition, K = 1, NPART.

Partition Size Card (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16	
Name	N				
Field	(1)	(2)	(3)	(4)	

N = The order of partition k

Elements in partition K of $[C_{h_o}]_j$. Format is given on the control card for $[C_{h_o}]_j$ matrix. All the elements in the AIC matrices are complex numbers, but the complexity is considered in the program. Thus, each partition may be input as though it is a real matrix of size N x 2N. The real elements form the odd number columns, and the imaginary elements in the even numbered columns.

For NFORM = 1 NROW = 1

Column	1-12	13-24	25-36	37-48	
Name	$a(Re)_{1,1}$	$a(I)_{1,1}$	$a(Re)_{1,2}$	$a(I)_{1,2}$	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card.

For NFORM = 1 and NROW = 0

Column	1-12	13-24	25-36	37-48	49-60	
Name	$a(Re)_{1,1}$	$a(I)_{1,1}$	$a(Re)_{2,1}$	$a(I)_{2,1}$		
Field	(1)	(2)	(3)	(4)	(5)	

Each column starts on a new card.

For NFORM = 0, NROW must = 0. Use column binary format. Column 1 starts in card Origin 1, Column 2 in location (1 + 2N), Column 3 in location (1 + 4N), etc. A TRA card must end each deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space required.

Flexible component AIC matrix $[C_h]_i$. The AIC matrices are often sparse; thus, a provision is made partitioning the matrix and entering only the nonzero partitions. The following data is repeated for i = 1, NSUR.

For $ISXT_i > 0$.

Control card for external stores partition of the surface i AIC matrix $[C_h]_{ij}$ (FORMAT 18I4)

Column	1-4	5-8	9-12	13-16
Name	NXST	(BLANK)	NFORM	NROW
Field	(1)	(2)	(3)	(4)

$NXST_i$ = Number of control points reserved for external stores

NFORM = 1 of the matrix elements are to be input using FORTRAN (FORMAT 6E12.8)

= 0 if the matrix elements are to be entered using column binary format

NROW = 1 if the matrix elements are to be entered by row

= 0 if the matrix elements are to be entered by column

Elements for external stores partition of AIC matrix $[C_h]_{ij}$

For NFORM = 1 and NROW = 1

Column	1-12	13-24	25-36	37-48
Name	$a(Re)_{1,1}$	$a(I)_{1,1}$	$a(Re)_{1,2}$	$a(I)_{1,2}$
Field	(1)	(2)	(3)	(4)

Each row begins on a new card.

For NFORM = 1 and NROW = 0

Column	1-12	13-24	25-36	37-48
Name	a(Re)1, 1	a(I)1, 1	a(Re)2, 1	a(Re)2, 1
Field	(1)	(2)	(3)	(4)

If NFORM = 0 then NROW must = 0 use column binary format. Column 1 starts in card origin 1, Column 2 in Location $(1+2NXT_i)$, Column 3 in Location $(1+4NXST_i)$, etc. A TRA Card must end each deck. The column binary format should be used only if the data are available as punched-card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space requirements.

Reference $1/k_i$ card for control point AIC matrix $\left[C_h\right]_{ij}$

FORMAT (6E12.8)

Column	1-12	13-24	25-36
Name	1/k		
Field	(1)	(2)	(3)

$1/k_i$ = The reduced velocity used in computing the flexible component $\left[C_h\right]_{ij}$. The AIC's must be computed for a $1/k_i$ which properly relates them to the jth $1/k_r$.

Control card for control point AIC matrix $\left[C_h\right]_{ij}$ (FORMAT (1814)

Column	1-4	5-8	9-12	13-16	
Name	NSIZE	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = Order of control point AIC matrix $\left[C_h\right]_{ij}$

NPART = Number of partitions in $\left[C_h\right]_{ij}$

NFORM = 1 matrix elements are to be input using FORTRAN (FORMAT 6E12.8)

= 0 matrix elements are to be input using column binary format

NROW = 1 elements input by row.

= 0 elements input by column.

Repeat the following data for j = 2, NPART. j = 1 corresponds to the external stores partition

Control card for partition size of partition j FORMAT (18J4)

Column	1-4	5-8	9-12	
Name	N			
Field	(1)	(2)	(3)	

N = Order of partition j

Elements of control point AIC matrix $\left[C_H\right]_{i,j}$ partition j

For NFORM = 1 and NROW = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	a(RE)1, 1	a(I)1, 1	a(Re)1, 2	a(I)1, 2	
Field	(1)	(2)	(3)	(4)	

Each row starts on a new card

For NFORM = 1 and NROW = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48	
Name	a(Re)1, 1	a(I)1, 1	a(Re)2, 1	a(I)2, 1	
Field	(1)	(2)	(3)	(4)	

Each column starts on a new card

For NFORM = 0 then NROW must = 0 use column binary format. Column 1 starts in card origin 1, Column 2 in Location (1 + 2N), Column 3 in Location (1 + 4N), etc. A TRA card must end each deck. The column binary format should be used only if the data are available as punched card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage space requirements

For $ISXT_i = 0$

Reference $1/k_i$ card for control point AIC matrix $\left[C_h\right]_{ij}$

FORMAT (6E12.8)

Column	1-12	13-24	25-36	
Name	$1/k_i$			
Field	(1)	(2)	(3)	

$1/k_i$ = The reduced velocity used in computing the flexible component $\left[C_h\right]_{ij}$. The AIC's must be computed for a $1/k_i$ which properly relates them to the jth $1/k_r$.

Control card for control point AIC matrix $\left[C_h\right]_{ij}$ FORMAT (18I4)

Column	1-4	5-8	9-12	13-16	
Name	NSIZE	NPART	NFORM	NROW	
Field	(1)	(2)	(3)	(4)	

NSIZE = Order of control point AIC matrix $\left[C_h\right]_{ij}$

NPART = Number of partitions in $\left[C_h\right]_{ij}$

NFORM = 1 matrix elements are to be input using FORTRAN (FORMAT 6E12.8)

= 0 matrix elements are to be input using column binary format.

NROW = 1 elements input by row

= 0 elements input by column

Repeat the following data for j = 1, NPART.

Control card for partition size of partition j FORMAT (18I4)

Column	1-4	5-8	9-12
Name	N		
Field	(1)	(2)	(3)

N = Order of partition

Elements of control point AIC matrix $\left[C_h\right]_{i,j}$ partition j

For NFORM = 1 and NROW = 1 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48
Name	a(Re)1, 1	a(I)1, 1	a(Re)1, 2	a(I)1, 2
Field	(1)	(2)	(3)	(4)

Each row starts on a new card

For NFORM = 1 and NROW = 0 (FORMAT 6E12.8)

Column	1-12	13-24	25-36	37-48
Name	a(Re)1, 1	a(I)1, 1	a(Re)2, 1	a(I)2, 1
Field	(1)	(2)	(3)	(4)

For NFORM = 0 then NROW must = 0 use column binary format. Column 1 start in card origin 1, Column 2 in Location (1 + 2N), Column 3 in Location (1 + 4N), etc. A TRA card must end each deck. The column binary format should be used only if the data are available as punched card output from appropriate computer programs. The only advantage of C-B format is the minimum card storage requirements.

SECTION 6

DESCRIPTION OF PROGRAM OUTPUT

A. Printed Output

1. All input data.
2. The dynamic matrix or flutter determinant if NPUNCH = -1 is used in the general control card.
3. For the vibration analysis or for each $1/k_r$ in a flutter analysis
   a. The eigenvalue for each output mode followed by the number of single- and double-precision iterations and the number of Aitken accelerations.
   b. The normalized eigenvectors (modes) followed by the check eigenvalues and eigenvectors.
   c. The frequencies (omegas) in cycles per second followed (in a flutter analysis) by the structural damping coefficient and the velocity (knots) assoicated with each frequency.
   d. If NPUNCH = ±1, the sequencing numbers used for identifying the punched-card output (frequencies and modes); this will conclude the printout for each $1/k_r$ used in a flutter analysis.
4. In a vibration analysis, the generalized mass corresponding to each output (free vibration) mode will follow the above printed output [see Eq. (28), Section I].
5. A number of different statements may be printed which indicate machine or program detected errors in input data (wrong format or incompatibility in the number of rows input for a matrix and the number designated).
6. If the program or the machine fails in the iteration portion of the program, a note will be printed stating the type or cause of failure.

7. If convergence is not obtained in the allowable number of iterations, a note will be printed and the program will continue. (In this case the eigenvalues and eigenvectors should be compared with the check eigenvalues and eigenvectors to determine if the convergence is sufficiently accurate.)

8. The printed output for the example problem is shown on the following pages.

SECTION 7

EXAMPLE PROBLEMS

As example problems, we choose a cantilever flutter analysis and a free-free symmetric flutter analysis of the jet transport wing (and rigid fuselage) treated throughout Ref. 5 and shown in Figure A-1. The wing mass, flexibility and aerodynamic matrices and the fuselage aerodynamic matrix (symmetric case) can be seen in the example problem printed output (pages 41-62) and will not be repeated with the list of input data to follow. The flexibility matrix normalizing factor is $K = 10^{-7}$. The wing weighting matrix is taken as unity ($[I]$) and, in this case, requires no input. The symmetrical rigid component (one-half of the fuselage) generalized mass matrix is given below in the pound-inch system

$$[\Delta m] = \begin{bmatrix} M_o & S_o \\ S_o & I_{yo} \end{bmatrix} = \begin{bmatrix} 17,400 & 1,370,250 \\ 1,370,250 & 4,457,907,200 \end{bmatrix}$$

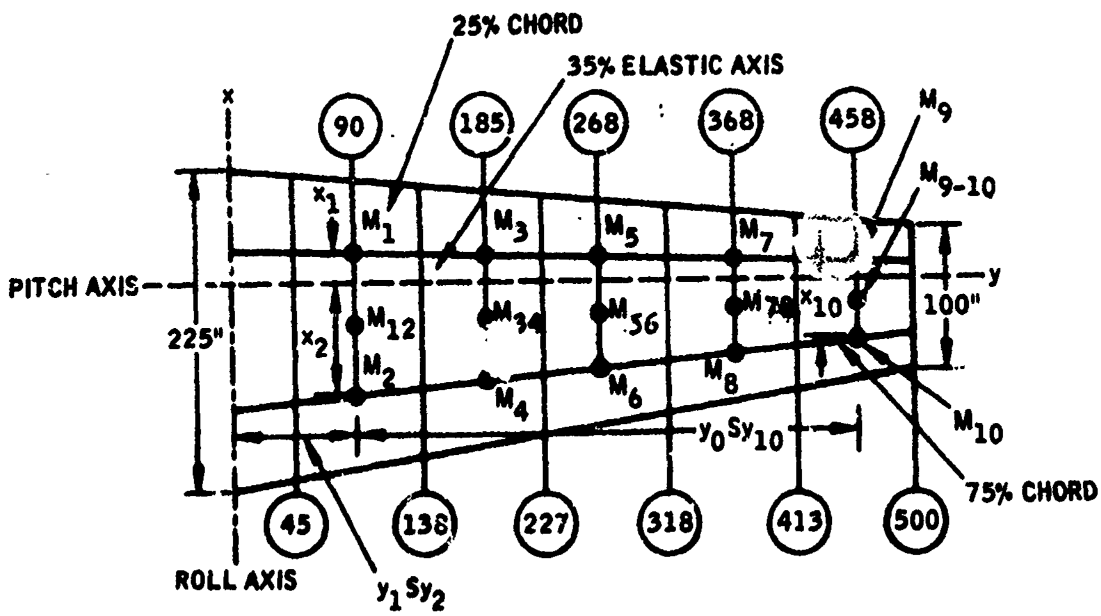


Figure A-1. Jet transport wing geometry.

In the antisymmetric case, the rigid component generalized mass matrix would be $[\Delta m] = [I_{xo}]$, and in the composite longitudinal-lateral case, the generalized mass matrix would be

$$[\Delta m] = \begin{bmatrix} M_o & S_o & 0 \\ S_o & I_{yo} & 0 \\ 0 & 0 & I_{xo} \end{bmatrix}$$

The symmetrical case requires the two rigid-body degrees of freedom of plunging and pitching. The rigid-body modal matrix, therefore, consists of two columns: a unit column, and a column of the x-coordinate of each control point. The rigid-body modal matrix for the wing is

$$[h_R] = [1 \quad x] = \begin{bmatrix} 1 & 20.25 \\ 1 & -81.00 \\ 1 & 17.85 \\ 1 & -71.40 \\ 1 & 15.80 \\ 1 & -63.20 \\ 1 & 13.30 \\ 1 & -53.20 \\ 1 & 11.05 \\ 1 & -44.20 \end{bmatrix}$$

The rigid-body modal matrix for the fuselage is

$$[h_{Ro}] = [1 \quad x] = \begin{bmatrix} 1 & -373.30 \\ 1 & -248.30 \\ 1 & -123.30 \\ 1 & +1.70 \\ 1 & +126.70 \end{bmatrix}$$

<u>Note</u> that the above matrix is used in computing the incremental generalized mass which results from the aerodynamic loads on one-half of the fuselage and, therefore, the x-coordinates must be given in the

proper order for the control points used in computing the fuselage AICs. [In this problem the fuselage AICs are hypothetical, but the x-coordinates are given in the order necessary for the slender-body theory AICs.]

The reference geometry for the wing is $b_{rw} = 5.468$ ft and $s_w = 37.917$ ft. The reference geometry for the fuselage is $b_{ro} = 5.468$ ft and $s_o = 18.9585$ ft. [We assume that the wing reference geometry was used to nondimensionalize the fuselage AIC matrix, and, because we require only one-half of the fuselage aerodynamic force, it can be obtained by setting $s_o = (1/2)s_w$.] Both example problems are carried out for the single reduced velocity $1/k_r = 16.67$ with the reference semichord for the system $b_r = 5.468$ ft and with sea-level density $\rho = 0.002378$ slugs/ft^3.

The output modes, the flutter frequencies and velocities, and the required structural damping can be seen in the example problem printed output (pages A-55 through A-65).

The input keypunch sheets are given, followed by the computer output.

AIRCRAFT

7094 FORTRAN CODING SHEET

NAME ______ PHONE ______ DATE ______ LOG NO. ______ PROB. NO. ______ PROG. NO. ______ PAGE ______ OF ______

```
                                    Title Card
    JET TRANSPORT EXAMPLE PROBLEM - CANTILEVER CASE                        HM100000
                                General Control Card
   1   1   0   0   1   4   0   0                                           HM100001
                Flexibility Matrix Normalizing Constant, K (FORMAT 6E12.8)
 1.0     -07                                                               HM100002
       ISXT_i and ISW_i (Controls for Number of External Stores Control Points and [W]_i, i = 1, ... NSUR)(FORMAT 18I4)
   0   0                                                                   HM100003
                  Reference b_r and Reference 1/k_r's (FORMAT 6E12.8)
 5.468       16.67                                                         HM100004
                Reference b_ri and Reference k_i for Each Surface (FORMAT 6E12.8)
 5.468      37.917                                                         HM100005
                         Density Series (FORMAT 6E12.8)
 0.002378                                                                  HM100006
                       NSIZE for Mass Matrix, [M] (FORMAT 18I4)
  10                                                                       HM100007
            Row Location of First and Last (LOW_r, HIGH)) Nonzero Element in Each Column of [M]
   1   2   1   2   3   4   3   4   5   6   5   6   7   8   7   8   9  10HM100008
```

HUGHES AIRCRAFT COMPANY

7094 FORTRAN CODING SHEET

PAGE ____ OF ____

NAME ____ PHONE ____ DATE ____ LOG NO. ____ PROB. NO ____ PROG NO ____

```
   9  10                                                                HM100009
```

[M] Matrix Elements (FORMAT 6E12.8)

```
 5383.6     -134.9                                                      HM100010
-  134.9      925.2                                                     HM100011
  20732.    -  11005.0                                                  HM100012
-11005.0     11478.                                                     HM100013
   3113.9    139.7                                                      HM100014
   139.7      806.6                                                     HM100015
 2638.8     -  21.                                                      HM100016
- 21.0        803.3                                                     HM100017
   487.5        7.3                                                     HM100018
 7.3          177.9                                                     HM100019
```

Control Card for Flexibility Matrix. [a] (FORMAT I614)

```
  10       1   0                                                        HM100020
```

[a] Matrix Elements

The matrix elements can be seen in the printed output. The control card gives the size and specifies the format (FORTRAN 6E12.8, by columns). See Section IV, Part B, Item 13 for other formats.

HUGHES AIRCRAFT COMPANY
7094 FORTRAN CODING SHEET

NAME ______ PHONE ______ DATE ______ LOG NO. ______ PROB. NO. ______ PROG. NO. ______ PAGE ______ OF ______

```
                        Reference 1/k for [C4] Matrix (FORMAT 6E12.8)
 16  67                                                                    HM100041
                        Control Card for [Ck] Matrix Input (FORMAT 18I4)
  10    5     1   1                                                        HM100042
          Partition k Size (FORMAT 18I4). Partition k Elements (FORMAT 6E12.8, by Row) k = 1, 5
   2                                                                       HM100043
 .300997  +03-60092673+02-30579615+03 32278209+02                          HM100044
 .1911816+00 16514119+02 5733 5448+00-16514119+02                          HM100045
   2                                                                       HM100046
 .29334980+03-55152775+02-29728241+03+31246942+02                          HM100047
 .14211353+00+13930788+02+42634058+00-13930788+02                          HM100048
   2                                                                       HM100049
 .30483998+03-53866061+02-30827594+03+31871354+02                          HM100050
 .11384775+00+12607991+02+34154829+00-12607991+02                          HM100051
   2                                                                       HM100052
 .32483251+03-62278329+02-32767460+03+32541166+02                          HM100053
 .84216232-01+11079562+02+25264879+00-11079562+02                          HM100054
   2                                                                       HM100055
```

6176CC REV. 2-64

HUGHES AIRCRAFT COMPANY

7094 FORTRAN CODING SHEET

NAME ______ PHONE ______ DATE ______ PAGE ______

```
 30321474+03-43827882+02-30521602+03+26517133+02                           HM100056
 53236897-01+84300262+01+15971068+00-84300262+01                           HM100057
 Other formats that can be used to input the [Cn] matrices are explained
 in Section III, Part B. Item 15.

      JET  TRANSPORT  EXAMPLE  PROBLEM  -  SYMMETRIC  CASE                  HM200000
   1    1     2     1     1    4     1     0                                HM200001
 K                                    (same as used for cantilever case)    HM200002
 ISXT ISW                             (same as used for cantilever case)    HM200003
 b,            1/k                    (same as used for cantilever case)    HM200004
               Reference br, and Reference ai for Each Surface  i = 0, 1, NSUR
 5.468         13.9585      6.468          37.917                           HM200005
 Density                              (same as used for cantilever case)    HM200006
                 Rigid Component Generalized Mass Matrix, [Am] (FORMAT 6E12.8, by Columns)
 17400.0       137025   +07                                                 HM200007
 137025    +07 4457907 2+10                                                 HM200008
 NSIZE for [M]                        (same as used for cantilever case)    HM200009
 LOW, HIGH                            (same as used for cantilever case)    HM200010
```

HUGHES AIRCRAFT CO.

7094 FORTRAN CODING SHEET

PAGE ____ OF ____

NAME ____ PHONE ____ DATE ____ LOG NO. ____ PROB NO. ____ PROG NO. ____

Line	Statement (columns 1–72)	Identification (columns 73–80)
1	LOW10 HIGH10 (same as used in cantilever case)	HM200011
2	[M] Matrix Elements (same as used in cantilever case)	
3	Number of Rows in Rigid Component Rigid-Body Modal Matrix, [h_{R_o}] (FORMAT 18I4)	
4	5	HM200022
5	[h_{R_o}] Matrix Elements (FORMAT 6E12.8, by Columns)	
6	1.0 1.0 1.0 1.0 1.0	HM200023
7	-373.30 -248.30 -123.30 +1.70 +126.70	HM200024
8	Number of Rows in Surface Rigid-Body Modal Matrix, [h_R] (FORMAT 18I4)	
9	10	HM200025
10	[h_R]$_1$ Matrix Elements (FORMAT 6E12.8, by Columns)	
11	1.0 1.0 1.0 1.0 1.0 1.0	HM200026
12	1.0 1.0 1.0 1.0	HM200027
13	20.25 -31.0 17.85 -71.40 15.80 -63.20	HM200028
14	13.30 -53.20 11.05 -44.20	HM200029
15	Control Card for [A] (same as used in cantilever case)	HM200030
16	[A] Matrix Elements (same as used in cantilever case)	
17	1/k for Rigid Component [C_{h_o}] (FORMAT 6E12.8)	
18	16.67	
19		
20		
21		
22		
23		
24		
25		

HUGHES AIRCRAFT COMPANY

7094 FORTRAN CODING SHEET

NAME ______ PHONE ______ DATE ______ LOG NO ______ PROB ______ PAGE ______

```
        Control Card for [Ch_h] Matrix Input (FORMAT 1314)                HM200052
   5    1    0    0                                                       
        [Ch_h] Matrix Elements (Column Binary Format)
   5
 The above control card indicates a 5 x 5 (unpartitioned) matrix to be input using column
 binary format.  The matrix can be seen in the printed output and will not be tabulated.
 (Other input formats are explained in Section II, Part B, item 15.)
        1/k for Surface [C_h] Matrix
 16.67                                                                    HM200064
        Control Card for Surface [C_h] Matrix Input
  10    5    1    1                                                       HM200065
        [C_h] Matrix by Partitions
 The partitions are identical to those used in the cantilever case.
```

JET TRANSPORT EXAMPLE PROBLEM - CANTILEVER CASE HM051641

FLUTTER ANALYSIS BY A COLLOCATION METHOD USING AERODYNAMIC INFLUENCE COEFFICIENTS

NSUR = 1 NAERO = 1 NRIGID = 0 NFUS = 0 NDENS = 1 MODES OUT = 4 NDELM = 0 NPUNCH = 0

B (REF) = 0.54680000E 01 K = 0.10000000E-06

SURFACE	B	S	EXTERNAL STORES SIZE
1	0.54580000E 01	0.37917000E 02	0

MASS MATRIX

SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.53836000E 04	-0.13490000E 03	0.	0.	0.	0.
2	-0.13490000E 03	0.92520000E 03	0.	0.	0.	0.
3	0.	0.	0.20732000E 05	-0.11005000E 05	0.	0.
4	0.	0.	-0.11005000E 05	0.11478000E 05	0.	0.
5	0.	0.	0.	0.	0.31139000E 04	0.13970000E 03
6	0.	0.	0.	0.	0.13970000E 03	0.88660000E 03
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

	COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN
1	0.	0.	0.	0.	
2	0.	0.	0.	0.	
3	0.	0.	0.	0.	
4	0.	0.	0.	0.	
5	0.	0.	0.	0.	
6	0.	0.	0.	0.	
7	0.26388000E 04	-0.21000000E 02	0.	0.	
8	-0.21000000E 02	0.88330000E 03	0.	0.	
9	0.	0.	0.48750000E 03	0.73000000E 01	
10	0.	0.	0.73000000E 01	0.17790000E 03	

FLEXIBILITY MATRIX

SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.87172000E 02	0.13361000E 02	0.12778000E 03	0.62720000E 02	0.16251000E 03	0.10492200E 03
2	0.13361000E 02	[illegible]	[illegible]	0.32297000E 03	0.10492200E 03	0.33529000E 03
3	0.12778000E 03	[illegible]	0.37732000E 03	0.15726000E 03	0.48255000E 03	0.37628000E 03
4	0.62720000E 02	[illegible]	0.15726000E 03	0.63749000E 03	0.37628000E 03	0.60135000E 03
5	0.16251000E 03	[illegible]	0.48255000E 03	0.37628000E 03	0.12758000E 04	0.11344300E 04
6	0.10492200E 03	[illegible]	0.37628000E 03	0.88136000E 03	0.11344300E 04	0.16999000E 04
7	0.20479000E 03	0.15634000E 03	0.73284000E 03	0.64338000E 03	0.19350000E 04	0.18169000E 04
8	0.15639000E 03	0.35021000E 03	0.64338000E 03	0.10012100E 04	0.18160000E 04	0.22920000E 04
9	0.24285000E 03	0.20257000E 03	0.95810000E 03	0.88378000E 03	0.25283000E 04	0.24294000E 04
10	0.23403000E 03	0.35785000E 03	0.88378000E 03	0.11810600E 04	0.24294000E 04	0.28249000E 04

	COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN
1	0.20478000E 03	0.15634000E 03	0.24285000E 03	0.23403000E 03	
2	0.15638000E 03	0.35021000E 03	0.20257000E 03	0.35785000E 03	
3	0.73284000E 03	0.64338000E 03	0.95810000E 03	0.88378000E 03	
4	0.64338000E 03	0.10012100E 04	0.88378000E 03	0.11810600E 04	
5	0.17350000E 04	0.18160000E 04	0.25283000E 04	0.[illegible]94000E 04	
6	0.18160000E 04	0.22920000E 04	0.24294000E 04	0.2[illegible]49000E 04	
7	0.35862000E 04	0.35052000E 04	0.52675000E 04	0.51171000E 04	
8	0.35052000E 04	0.42290000E 04	0.51171000E 04	0.57187000E 04	
9	0.52675000E 04	0.51171000E 04	0.64840000E 04	0.82340000E 04	
10	0.51171000E 04	0.57187000E 04	0.82340000E 04	0.92310000E 04	

WEIGHTING MATRIX

SURFACE 1, NO WEIGHTING MATRIX

AERODYNAMIC MATRIX 1./K R = 0.1000000E 02
SURFACE 1. 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.36099700E 03	-0.68092673E 02	-0.30579615E 03	0.32278789E 02	0.	0.
2	0.19111816E 00	0.16514119E 02	0.57335446E 00	-0.16514119E 02	0.	0.
3	0.	0.	0.	0.	0.29334980E 03	-0.55152775E 02
4	0.	0.	0.	0.	0.14211353E 00	0.13930788E 02
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

	COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN 11	COLUMN 12
1	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.
3	-0.29728241E 03	0.31246942E 02	0.	0.	0.	0.
4	0.42634856E 00	-0.13934788E 02	0.	0.	0.	0.
5	0.	0.	0.38483998E 03	-0.53868861E 02	-0.38827594E 03	0.51871554E 02
6	0.	0.	0.11384775E 00	0.12687991E 02	0.34154329E 00	-0.12687991E 02
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

	COLUMN 13	COLUMN 14	COLUMN 15	COLUMN 16	COLUMN 17	COLUMN 18
1	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.32483[illegible]51E 03	-0.52278329E 02	-0.32767[illegible]E 03	0.32541166E 02	0.	0.
8	0.84215[illegible]E-01	0.11079562E 02	0.25264[illegible]70E 00	-0.11079562E 02	0.	0.
9	0.	0.	0.	0.	0.30321474E 03	-0.43827862E 02
10	0.	0.	0.	0.	0.53236897E-01	0.84300262E 01

	COLUMN 19	COLUMN 20	COLUMN
1	0.	0.	
2	0.	0.	
3	0.	0.	
4	0.	0.	
5	0.	0.	
6	0.	0.	
7	0.	0.	
8	0.	0.	
9	-0.[illegible]E 03	[illegible]3E 02	
10	[illegible]	[illegible]E 01	

OUTPUT DATA

FLUTTER ANALYSIS BY A COLLOCATION METHOD, USING AERODYNAMIC INFLUENCE COEFFICIENTS

DENSITY = 0.23780000E-02   REDUCED VELOCITY = 0.16670000E 02

RIGID BODY DEGREES OF FREEDOM

MODE	EIGENVALUE		ITERATIONS S.P.	D.P.	AITKENS S.P.	D.P.
1	0.28514254E 08	-0.25182939E 08	10	0	2	0
2	0.17153621E 08	-0.28573745E 08	18	0	4	0
3	0.63295259E 07	0.11860048E 07	16	0	4	0
4	0.28438861E 07	-0.12622640E 07	20	0	4	3

EIGENVECTORS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.41765569E-01	0.39935716E-02	0.57422154E-01	0.27973628E-01	0.11260764E 00	-0.38122170E-01
2	0.33611827E-01	0.67522708E-02	0.60325008E-01	-0.27717133E-02	0.95814571E-01	-0.47620138E-01
3	0.14070067E 00	0.81289036E-02	0.17850658E 00	0.50252555E-01	0.26814691E 00	-0.69573386E-01
4	0.12642782E 00	0.13288543E-01	0.17790778E 00	-0.14084830E-02	0.24426274E 00	-0.60556956E-01
5	0.36103484E 00	0.17490047E-01	0.42532389E 00	0.86469743E-01	0.58779018E 00	-0.14839706E 00
6	0.34535172E 00	0.25502904E-01	0.44030061E 00	0.33060263E-01	0.59176531E 00	-0.17162145E 00
7	0.67711089E 00	0.14759703E-01	0.71753938E 00	0.68290335E-01	0.82465572E 00	-0.81978349E-01
8	0.65968206E 00	0.26878921E-01	0.74431981E 00	0.29982516E-01	0.86821067E 00	-0.13225259E 00
9	0.10000000E 01	0.57693956E-08	0.96504349E 00	0.27935114E-01	0.93070006E 00	0.65759583E-01
10	0.98173522E 00	0.14738217E-01	0.10000000E 01	0.71289627E-10	0.10000000E 01	0.14332015E-09

	COLUMN 7	COLUMN 8	COLUMN
1	-0.85581770E-01	-0.11844954E 00	
2	-0.14422336E 00	-0.92387966E-01	
3	-0.15972682E 00	-0.17172985E 00	
4	-0.16794566E 00	-0.157.0267E 00	
5	-0.46918808E 00	-0.17309963E 00	
6	-0.45836876E 00	-0.26935221E 00	
7	-0.13126460E-01	-0.66963119E-01	
8	0.97545000E-0[illegible]	[illegible]31642 9E 00	
9	0.10000000E 01	[illegible]	
10	0.95827440E 01	[illegible]46321 1E-01	

CHECK EIGENVALUES AND EIGENVECTORS

0.28514267E 08 -0.25183840E 05 0.12153636E 08 -0.28584727E 06 0.63295459E 07 0.11864946E 07
0.20439634E 07 -0.12623659E 07

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.41765565E-01	0.59935390E-02	0.57421844E-01	0.27973633E-01	0.11260817E 00	-0.38121185E-01
2	0.33611786E-01	0.67527255E-02	0.60324983E-01	-0.27713930E-02	0.95815358E-01	-0.47126920E-01
3	0.14070066E 00	0.81288441E-02	0.17058602E 00	0.50252573E-01	0.26811789E 00	-0.69571621E-01
4	0.12642694E 00	0.13288569E-01	0.17798768E 00	-0.13998269E-02	0.24426423E 00	-0.90548721E-01
5	0.36103078E 00	0.17496553E-01	0.42532286E 00	0.88469934E-01	6.58779208E 00	-0.14839379E 00
6	0.34535159E 00	0.25592861E-01	0.44038804E 00	0.33061094E-01	0.59176783E 00	-0.17164619E 00
7	0.67711881E 00	0.14759617E-01	0.71753857E 00	0.68293261E-01	0.82485651E 00	0.81976559E-03
8	0.65968186E 00	0.27870910E-01	0.74431851E 00	0.29903183E-01	0.86821284E 00	-0.13224914E 00
9	0.10000000E 01	0.19363443E-08	0.96504330E 00	0.27934334E-01	0.93069882E 00	0.65757559E-01
10	0.98173518E 00	0.14733288E-01	0.10000000E 01	0.19184554E-09	0.10000000E 01	0.48020418E-09

	COLUMN 7	COLUMN 8	COLUMN
1	-0.85493178E-01	-0.11845119E 00	
2	-0.10421509E 00	-0.97384766E-01	
3	-0.15970453E 00	-0.1[illegible]72668E 00	
4	-0.18188963E 00	-0.[illegible]716879E 00	
5	-0.46907590E 00	-0.[illegible]309738E 00	
6	-0.4583351[illegible]E 00	-0.20537629E 00	
7	-0.18102238E-01	-0.66989833E-01	
8	0.97804460E-02	-0.12320435E 00	
9	0.10000000E 01	0.81157384E-09	
10	0.95823531E 00	-0.11462694E-01	

MODE	FRQCY (CPS)	DAMPING	VELOCITY (KNOTS)
1	0.18517534E 01	-0.88317019E 00	0.62794529E 03
2	0.28363596E 01	-0.23510478E-01	0.96183365E 03
3	0.39303258E 01	0.18747073E 00	0.13328069E 04
4	0.69164967E 01	-0.61758042E 00	0.23454407E 04

JET TRANSPORT EXAMPLE PROBLEM - SYMMETRIC CASE   MM051699
FLUTTER ANALYSIS BY A COLLOCATION METHOD USING AERODYNAMIC INFLUENCE COEFFICIENTS

NSUR = 1 NAERO = 1 NRIGID = 2 NFUS = 1 NDENS = 1 MODES OUT = 4 NDELM = 1 NPUNCH = 0

B RIGID COMPONENT = 0.54680000E 01   S RIGID COMPONENT = 0.16958500E 02

B (REF) = 0.54680000E 01   K = 0.10000000E-06

SURFACE	B	S	EXTERNAL STORES SIZE
1	0.54680000E 01	0.37917000E 02	0

RIGID COMPONENT MASS MATRIX

	COLUMN 1	COLUMN 2	COLUMN
1	0.17400000E 05	0.13702500E 07	
2	0.13702500E 07	0.44579072E 10	

MASS MATRIX
SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.53836000E 04	-0.13490000E 03	1.	0.	0.	0.
2	-0.13490000E 03	0.92520000E 03	0.	0.	0.	0.
3	0.	0.	0.20732000E 05	-0.11005000E 05	0.	0.
4	0.	0.	-0.11005000E 05	0.11478000E 05	0.	0.
5	0.	0.	0.	0.	0.31139000E 04	0.139[illegible]E 03
6	0.	0.	0.	0.	0.13970000E 03	0.80[illegible]E 03
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

	COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN
1	0.	0.	0.	0.	
2	0.	0.	0.	0.	
3	0.	0.	0.	0.	
4	0.	0.	0.	0.	
5	0.	0.	1.	0.	
6	0.	0.	1.	0.	
7	0.26388000E 04	-0.21000000E 02	0.	0.	
8	-0.21000000E 02	0.80330000E 03	0.	0.	
9	0.	0.	0.48750000E 03	0.73000000E 01	
10	0.	0.	0.73000000E 01	0.17790000E 03	

RIGID COMPONENT MODES,   2 CONTROL POINTS.

	COLUMN 1	COLUMN 2	COLUMN
1	0.10000000E 01	-0.37330000E 03	
2	0.10000000E 01	-0.24830000E 03	
3	0.10000000E 01	[illegible]	
4	0.10000000E 01	0.3[illegible]E 03	
5	0.10000000E [illegible]	0.1267000E 03	

RIGID BODY MODAL MATRIX

SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN
1	0.10000000E 01	0.2025[illegible]E 02	
2	0.10000000E 01	-0.6[illegible]000000E 02	
3	0.10000000E 01	0.17850000E 02	
4	0.10000000E 01	-0.71420000E 02	
5	0.10000000E 01	0.15800000E 02	
6	0.10000000E 01	-0.63200000E 02	
7	0.10000000E 01	0.13360000E 02	
8	0.10000000E 01	-0.53200000E 02	
9	0.10000000E 01	0.11050000E 02	
10	0.10000000E 01	-0.44200000E 02	

FLEXIBILITY MATRIX

SURFACE 1, 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.87172600E 02	0.13361000E 02	0.12778000E 03	0.62720000E 02	0.16251000E 03	0.10492200E 03
2	0.13361000E 02	0.30861000E 03	0.62720000E 02	0.32297000E 03	0.10492200E 03	0.33529800E 03
3	0.12779800E 03	0.62721900E 02	0.27732000E 03	0.15726000E 03	0.48255000E 03	0.37628000E 03
4	0.62720000E 02	0.32297000E 03	0.15726000E 03	0.63749000E 03	0.37628000E 03	0.80136000E 03
5	0.16251000E 03	0.10492200E 03	0.48255000E 03	0.37628000E 03	0.12758000E 04	0.11344300E 04
6	0.10492200E 03	0.33529800E 03	0.37628000E 03	0.80136000E 03	0.11344300E 04	0.16999000E 04
7	0.20478000E 03	0.15630000E 03	0.73284000E 03	0.64338000E 03	0.19350000E 04	0.18160000E 04
8	0.15630000E 03	0.35021000E 03	0.64338000E 03	0.10012100E 04	0.18160000E 04	0.22920000E 04
9	0.24285000E 03	0.20257000E 03	0.95810000E 03	0.88378000E 03	0.25283000E 04	0.24294000E 04
10	0.20403000E 03	0.35785000E 03	0.88378000E 03	0.11810600E 04	0.24294000E 04	0.28249000E 04

	COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN
1	0.20478000E 03	0.15630000E 03	0.24285000E 03	0.20403000E 03	
2	0.15630000E 03	0.35021000E 03	0.20257000E 03	0.35785000E 03	
3	0.73284000E 03	0.64338000E 03	0.95810000E 03	0.88378000E 03	
4	0.64338000E 03	0.10012100E 04	0.88378000E 03	0.11810600E 04	
5	0.19350000E 04	0.18160000E 04	0.25283000E 04	0.24294000E 04	
6	0.18160000E 04	0.22920000E 04	0.24294000E 04	0.28249000E 04	
7	0.36862000E 04	0.35052000E 04	0.52675000E 04	0.51171000E 04	
8	0.35052000E 04	0.42292000E 04	0.51171000E 04	0.57187000E 04	
9	0.52675000E 04	0.51171000E 04	0.84840000E 04	0.82340000E 04	
10	0.51171000E 04	0.57187000E 04	0.82340000E 04	0.92340000E 04	

WEIGHTING MATRIX

SURFACE 1, NO WEIGHTING MATRIX

RIGID COMPONENT AERO MATRIX. 5 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	-0.19883166E 01	-0.71028437E 01	-0.31782872E 01	0.99733821E 01	0.17128096E 01	-0.24681993E 01
2	0.20347044E 01	-0.36570088E 01	-0.89710928E 01	0.67589525E 00	0.25888789E 01	0.32829072E 01
3	0.	0.	0.25888789E 01	-0.32829072E 01	-0.89710928E 01	-0.67589525E 00
4	0.	0.	0.	0.	0.18608618E 01	-0.23997266E 01
5	0.	0.	0.	0.	0.19459838E 00	0.73566824E 00

	COLUMN 7	COLUMN	COLUMN 9	COLUMN 10	COLUMN
1	0.	0.	0.	0.	
2	0.	0.	0.	0.	
3	0.20347044E 01	0.30575088E 01	0.	0.	
4	-0.59000403E 01	-0.47448609E 00	0.10709520E 01	0.17538263E 01	
5	0.45455574E 00	-0.31600294E 01	-0.19732837E 01	0.81508698E 00	

AERODYNAMIC MATRIX 1./K R = 0.16670000E 02

SURFACE 1. 10 CONTROL POINTS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.38099700E 03	-0.68092673E 02	-0.30579615E 03	0.32278289E 02	0.	0.
2	0.19111816E 00	0.16514119E 02	0.57335448E 00	-0.16514119E 02	0.	0.
3	0.	0.	0.	0.	0.29334980E 03	-0.55152775E 02
4	0.	0.	0.	0.	0.14211353E 00	0.13930788E 02
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

	COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN 11	COLUMN 12
1	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.
3	-0.29728241E 03	0.31246962E 02	0.	0.	0.	0.
4	0.42434056E 00	-0.13937288E 02	0.	0.	0.	0.
5	0.	0.	0.30383990E 03	-0.53868861E 02	-0.30827594E 03	0.31871554E 02
6	0.	0.	0.13384775E 00	0.12607991E 02	0.34154329E 00	-0.12607991E 02
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.

	COLUMN 13	COLUMN 14	COLUMN 15	COLUMN 16	COLUMN 17	COLUMN 18
1	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.32483251E 03	-0.52278329E 02	-0.32757460E 03	0.37541166E 02	0.	0.
8	0.84216232E-01	0.11079562E 02	0.25264870E 00	-0.11079562E 02	0.	0.
9	0.	0.	0.	0.	0.30321474E 03	-0.43827882E 02
10	0.	0.	0.	0.	0.53236897E-01	0.64300262E 01

	COLUMN 19	COLUMN 20	COLUMN
1	0.	0.	
2	0.	0.	
3	0.	0.	
4	0.	0.	
5	0.	0.	
6	0.	0.	
7	0.	0.	
8	0.	0.	
9	-0.30521062E 03	0.28517173E 02	
10	0.15971048E 00	-0.64300262E 01	

OUTPUT DATA

FLUTTER ANALYSIS BY A COLLOCATION METHOD, USING AERODYNAMIC INFLUENCE COEFFICIENTS

DENSITY = 0.23780000E-02 REDUCED VELOCITY = 0.16670000E 02

RIGID BODY DEGREES OF FREEDOM

MODE	EIGENVALUE		ITERATIONS S.P.	D.P.	AITKENS S.P.	D.P.
1	0.22933607E 00	-0.15925949E 00	13	0	3	0
2	0.13014712E 00	-0.15691301E 06	15	0	4	0
3	0.48985828E 07	0.13604431E 07	13	0	2	0
4	0.19962581E 07	-0.12316837E 07	18	0	4	6

EIGENVECTORS

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	-0.22672200E 00	0.92881241E-01	-0.10988617E 00	-0.10249931E 00	-0.11566342E 00	0.10828242E-02
2	-0.24322614E 00	0.96231452E-01	-0.11257997E 00	-0.13723300E 00	-0.14202341E 00	0.42084159E-03
3	-0.10896546E 00	0.96128553E-01	0.36834285E-02	-0.72438926E-01	0.39344759E-01	-0.37217868E-01
4	-0.13140213E 00	0.95914406E-01	0.38691290E-02	-0.12856633E 00	0.62680741E-02	-0.41506405E-01
5	0.16390737E 00	0.94669461E-01	0.27384991E 00	-0.15872387E-01	0.38962762E 00	-0.14034653E 00
6	0.13893196E 00	0.93805569E-01	0.27711217E 00	-0.70519944E-01	0.38137447E 00	-0.15814607E 00
7	0.57352658E 00	0.51462690E-01	0.65861424E 00	0.20940761E-01	0.72148682E 00	-0.93107597E-01
8	0.54571533E 00	0.66242830E-01	0.64748616E 00	-0.26460319E-01	0.75147159E 00	-0.13747385E 00
9	0.10000000E 01	0.39256811E-08	0.98694071E 00	0.36698160E-01	0.94384034E 00	0.67121342E-01
10	0.97106495E 00	0.19206324E-03	0.10000000E 01	0.18268105E-09	0.10000000E 01	0.73126528E-09

	COLUMN 7	COLUMN 8	COLUMN
1	0.55517504E-01	-0.28637167E-01	
2	0.37303378E-03	-0.33456806E-02	
3	-0.61298371E-01	-0.81922612E-01	
4	-0.72267180E-01	-0.58644277E-01	
5	-0.53630956E 00	-0.13743654E 00	
6	-0.51810123E 00	-0.16383449E 00	
7	-0.12079909E 00	-0.69967863E-01	
8	-0.82737129E-01	-0.1[illegible]E 00	
9	0.10000000E 01	0.[illegible]	
10	0.94309171E 00	-0.[illegible]7047.6E-02	

CHECK EIGENVALUES AND EIGENVECTORS

0.22933607E 08 -0.15925953E 08 0.13014710E 08 -0.15691698E 06 0.48905000E 07 0.13604393E 07
0.19962599E 07 -0.12316878E 07

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	-0.22672210E 00	0.97801248E-01	-0.16930607E 00	-0.10249928E 00	-0.11566328E 00	0.10829196E-02
2	-0.24372616E 00	0.96231465E-01	-0.11252989E 00	-0.13723299E 00	-0.14202321E 00	0.42007173E-03
3	-0.10896548E 00	0.90125561E-01	0.36834970E-02	-0.72438898E-01	0.39344850E-01	-0.37217773E-01
4	-0.13140214E 00	0.95914421E-01	0.38692318E-02	-0.12866630E 00	0.62682456E-02	-0.41506375E-01
5	0.16390735E 00	0.84669469E-01	0.27384996E 00	-0.15872351E-01	0.38962777E 00	-0.14034850E 00
6	0.13893196E 00	0.93805592E-01	0.27711225E 00	-0.70519987E-01	0.38137466E 00	-0.15814662E 00
7	0.57352656E 00	0.51462696E-01	0.63801430E 00	0.20940764E-01	0.72148695E 00	-0.93109646E-01
8	0.54571[illegible]E 00	0.66247832E-01	0.64748619E 00	-0.26460305E-01	0.75147167E 00	-0.13747361E 00
9	0.10000000E 01	0.18256829E-08	0.86694071E 00	0.36698161E-01	0.94384057E 00	0.67121243E-01
10	0.97106489E 00	0.19260280E-01	0.10000000E 01	0.19268157E-10	0.10000000E 01	0.11595552E-06

	COLUMN 7	COLUMN 8	COLUMN
1	0.55517193E-03	-0.28630435E-01	
2	0.37502715E-01	-0.33451963E-02	
3	-0.61298424E-01	-0.81924672E-01	
4	-0.72267375E-01	-0.58643527E-01	
5	-0.53639659E 00	-0.13743543E 00	
6	-0.51[illegible]036E 00	-0.16303335E 00	
7	-0.12079819E 00	-0.69967353E-01	
8	-0.82736019E-01	-0.11890159E 00	
9	0.10000000E 01	0.12743769E-08	
10	0.94399217E 00	-0.80786561E-02	

MODE	FRQCY (CPS)	DAMPING	VELOCITY (KNOTS)
1	0.20447998E 01	-0.69443715E 00	0.70819116E 03
2	0.27499231E 01	-0.12056507E-01	0.92947031E 03
3	0.44713368E 01	0.27818862E 00	0.15162683E 04
4	0.69985134E 01	-0.61699621E 00	0.23732553E 04

SECTION 8

PROGRAM LISTING

```
$      OPTION  FORTRAN
$      FORTRAN LSTOU,DECK
C MAIN       FLUTTER OVERLAY
C            JAN 13,1967
      COMMON IT(218)
C     CHANGE  BCD TAPES TO BINARY TAPES
   10 CALL LLINK (6HPAR1TT)
      CALL PART1
      CALL LLINK (6HPAR2TT)

      CALL PART2
      GO TO 10
      END
$      FORTRAN LSTOU,DECK
C MPRINT
      SUBROUTINE MPRINT (A,M,N,MD,NTAPE)                                HM14007
      DIMENSION   A(1), IT(6), C(6)                                     HM14007
      EQUIVALENCE  (IT,C)                                               HM14007
    2 FORMAT    (1H , 4X, 6( 6X, 7HCOLUMN   1I4  )   ///     )          HM14008
    3 FORMAT    (1H  1I4, X,    (6E 17.8)  )                            HM14008
      N1=N                                                              HM14008
      N2=6                                                              HM14008
      N3=6                                                              HM14008
      N4=1                                                              HM14008
    4 IF (N3-N1)  6,6,5                                                 HM14008
    5 N2=N1-N3+6                                                        HM14008
      N3=N1                                                             HM14009
    6 K=0                                                               HM14009
      DO  7  I= N4,N3                                                   HM14009
        K=K+1                                                           HM14009
    7   IT(K)=I                                                         HM14009
       WRITE (NTAPE,2) (IT(I),I=1,N2)
      DO  9  I=1,M                                                      HM14009
        K=0                                                             HM14009
        L=MD*(N4-1)+I                                                   HM14009
        DO  8  J=N4,N3                                                  HM14009
          K=K+1                                                         HM14010
          C(K)=A(L)                                                     HM14010
    8     L=L+MD                                                        HM14010
    9  WRITE (NTAPE,3) I,(C(K),K=1,N2)
      IF (N3-N1) 10,11,11                                               HM14010
   10 N3=N3+6                                                           HM14010
      N4=N4+6                                                           HM14010
      GOTO  4                                                           HM14010
   11 RETURN                                                            HM14010
      END                                                               HM14010
$      FORTRAN LSTOU,DECK
C MPUNCH
      SUBROUTINE MPUNCH(A,M,N,IOUT,ITRA,IORG,BCDZ,MAXM,NTAPE,NCARDS)
      DIMENSION A(1)
      RETURN
      END
$      LINK    PAR1TT
$      FORTRAN LSTOU,DECK
C PART1
      SUBROUTINE PART1
C     NSUR=TOTAL NUMBER OF SURFACES ALLOWED.
C     NDENS=TOTAL NUMBER OF DENSITIES ALLOWED.
C     NRIGID=TOTAL NUMBER OF RIGID BODIES ALLOWED
C     NSIZE=TOTAL NUMBER CONTROL POINTS ON ANY ONE SURFACE ALLOWED
C     NMODES=TOTAL NUMBER MODES INPUT ON ANY ONE SURFACE.
```

```
      DIMENSION  ISXST(20), ISH(20), BR(21), S(21), RHO(20), DM(6,6),
     1           DD(6,12), DMBAR(6,12), BARMRR(6,12), LOW(50), LHIGH(50)
     2           , IT(218), VELCTY(20), NSIZES(20),
     3           A(50,100), F(50,100), U(50,100), HR(50,6), HRT(6,100),
     4           HT(6,100), G(6,100)
C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO
C   COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---
      COMMON  IT
      EQUIVALENCE  (IT(1),ISXST), (IT(21),ISH), (IT(41),RHO) ,
     1            ,(IT(61),NTAPE1), (IT(62),NTAPE2), (IT(63),NTAPE3),
     2             (IT(64),NTAPE4), (IT(65),NTAPE5), (IT(66),NTAPE6),
     3             (IT(67),NTAPE7), (IT(68),NTAPE8), (IT(69),NTAPE9),
     4             (IT(70),NSUR), (IT(71),NRIGID), (IT(72),BREF),
     5             (IT(73),NAERO), (IT(74),NFUS), (IT(75),NDENS),
     6             (IT(76),MODES), (IT(77),NPOINT), (IT(78),NPUNCH),
     7             (IT(79),MAXR), (IT(80),MAXQ), (IT(81),MAXS),
     8             (IT(82),NC), (IT(83),NSURFS), (IT(84),NR2),
     9             (IT(85),BR), (IT(106),S), (IT(127),VELCTY)
      EQUIVALENCE  (IT(147),NTAPEU), (IT(148),NRHO), (IT(149),NITRSP),
     1             (IT(150),NITRDP), (IT(151),EPSP), (IT(152),EPDP),
     2             (IT(153),PICON), (IT(154),NVEL), (IT(155),NCARDS),
     3             (IT(156),AITKEN), (IT(157),NGO), (F,G), (F(601,1),HT)
     4,              (U,LOW), (U(51,1),LHIGH), (IT(158),FK), (IT(159),SQR
     5CON)           , (IT(160),NSIZES), (IT(180),DM), (DMBAR,BARMRR)
     6               , (IT(216),NDELM),(IT(217),AITKED),(IT(218),KPART)
    1 FORMAT  (18I4)
    2 FORMAT  (6E12.8)
    3 FORMAT  (1H 16X, 41H FLUTTER ANALYSIS BY A COLLOCATION METHOD
     1        42H USING AERODYNAMIC INFLUENCE COEFFICIENTS ///10H NSUR =
     2         I12, 10H  NAERO =  I14, 11H  NRIGID = I12, 9H  NFUS =
     3         I12, 10H  NDENS =  I14, 14H  MODES OUT =  I12
     4         , 9H NDELM = I12,10H NPUNCH =  I12    )
    4 FORMAT (1H0 22X, 11H B (REF) = 1E20.8, 5X, 4HK = 1E20.8 /1H0 25X,
     1        7HSURFACE 18X, 1HB 19X, 1HS 10X,20HEXTERNAL STORES SIZE //)
    5 FORMAT  (1H0 10X, 21H B RIGID COMPONENT =  1E18.8, 5X, 8H S RIGID
     1         13H COMPONENT =  1E18.8  )
    6 FORMAT  (1H  I129,  2(  5X,  1E20.8),  I112 )
   10 FORMAT  (1H1  48X, 12H MASS MATRIX     )
   11 FORMAT  (41H0 NUMBER OF CONTROL POINTS THIS MATRIX, ( I14,
     1         48H) AND TOTAL NUMBER OF CONTROL POINTS EXPECTED, (  I14,
     2         37H) DO NOT AGREE. PROGRAM CONTINUED....   )
   12 FORMAT  (1H1 42X,24H RIGID BODY MODAL MATRIX   )
   13 FORMAT  (1H  38X,  8HSURFACE  I12,  1H,  I16, 15H CONTROL POINTS)
   14 FORMAT  (1H1 43X, 20H FLEXIBILITY MATRIX     )
   15 FORMAT  (1H1  46X, 18H WEIGHTING MATRIX    )
   16 FORMAT  (1H1 28X, 29H RIGID COMPONENT AERO MATRIX, I19, 8H CONTROL
     1         7H POINTS      )
   17 FORMAT  (1H1   33X,  20H AERODYNAMIC MATRIX  8X, 10H 1./K R =
     1          1E20.8   )
   18 FORMAT  (1H0 30X, 23H RIGID COMPONENT MODES,  I110,
     1         17H CONTROL POINTS.     )
   19 FORMAT  (1H0 30X, 29H RIGID COMPONENT MASS MATRIX     )
   20 FORMAT  (47H0 ERROR IN INVERSE ROUTINE. PROGRAM TERMINATED     )
   21 FORMAT  ( 1H   38X, 8HSURFACE  I12,  1H, 6X, 13H NO WEIGHTING
     1          7H MATRIX  )
   22 FORMAT  ( 3E12.8, 2I4 )
   23 FORMAT  ( 8H0EPSP =    1E16.8, 1X, 8H EPDP =  1E16.8, 1X,
     1         10H AITKEN =  1E16.8, 1X, 10H NITRSP = I14, 1X,
     2         10H NITRDP =    I14  )
   24 FORMAT  (1H0 43X, 20H GENERALIZED MASSES  ////    (1H  30X,
     1          5H MASS  I14,  3H =   1E16.8 )  )
```

```
   25 FORMAT   ( 1H0   30X, 23H RIGID COMPONENT MODES  ///     )
   26 FORMAT   ( 2I4, 62X, 1A6, 1I4 )
   27 FORMAT   ( 1H0   33X, 23H PUNCHED CARDS NUMBERS  1A6, 1I4, 6H THRU
     1                1A6, 1I4   )
   28 FORMAT (1H0   20X,   3H K= 1E16.8     )
      DATA Q000CT/020203040440005/
      BCDZ =Q000CT


$       OPTION   FORTRAN
$       FORTRAN LSTOU.DECK
C MAIN       FLUTTER OVERLAY
C            JAN 13,1967
      COMMON IT(218)
C     CHANGE  BCD TAPES TO BINARY TAPES
   10 CALL LLINK (6HPAR1TT)
      CALL PART1
      CALL LLINK (6HPAR2TT)
```

```
$       OPTION  FORTRAN
$       FORTRAN LSTOU.DECK
C MAIN        FLUTTER OVERLAY
C             JAN 13,1967
      COMMON IT(218)
C     CHANGE  BCD TAPES TO BINARY TAPES
   10 CALL LLINK (6HPAR1TT)
      CALL PART1
      CALL LLINK (6HPAR2TT)

      CALL PART2
      GO TO 10
      END
$       FORTRAN LSTOU.DECK
C MPRINT
      SUBROUTINE MPRINT (A,M,N,MD,NTAPE)
      DIMENSION   A(1), IT(6), C(6)
      EQUIVALENCE  (IT,C)
    2 FORMAT   (1H , 4X, 6( 6X, 7HCOLUMN   1I4  )   ///    )
    3 FORMAT   (1H  1I4, X,   (6E 17.8)  )
      N1=N
      N2=6
      N3=6
      N4=1
    4 IF (N3-N1)  6,6,5
    5 N2=N1-N3+6
      N3=N1
    6 K=0
      DO  7  I= N4,N3
        K=K+1
    7   IT(K)=I
      WRITE (NTAPE,2) (IT(I),I=1,N2)
      DO  9  I=1,M
        K=0
        L=MD*(N4-1)+I
        DO  8  J=N4,N3
          K=K+1
          C(K)=A(L)
    8     L=L+MD
    9  WRITE (NTAPE,3) I,(C(K),K=1,N2)
       IF (N3-N1) 10,11,11
   10  N3=N3+6
       N4=N4+6
       GOTO  4
   11  RETURN
       END
$       FORTRAN LSTOU.DECK
C MPUNCH
       SUBROUTINE MPUNCH(A,M,N,IOUT,ITRA,IORG,BCDZ,MAXM,NTAPE,NCARDS)
       DIMENSION A(1)
       RETURN
       END
$       LINK    PAR1TT
$       FORTRAN LSTOU.DECK
C PART1
       SUBROUTINE PART1
C      NSUR=TOTAL NUMBER OF SURFACES ALLOWED.
C      NDENS=TOTAL NUMBER OF DENSITIES ALLOWED.
C      NRIGID=TOTAL NUMBER OF RIGID BODIES ALLOWED
C      NSIZE=TOTAL NUMBER CONTROL POINTS ON ANY ONE SURFACE ALLOWED
C      NMODES=TOTAL NUMBER MODES INPUT ON ANY ONE SURFACE.
```

```
      DIMENSION  ISXST(20), ISH(20), HR(21), S(21), RHO(20), DM(6,6),
     1           DO(6,12), DMBAR(6,12), BARMRR(6,12), LOW(50), LHIGH(50)
     2           , IT(218), VELCTY(20), NSIZES(20),
     3           A(50,100), F(50,100), U(50,100), HR(50,6), IRT(6,100),
     4           HT(6,100), G(6,100)
C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO
C   COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---
      COMMON  IT
      EQUIVALENCE  (IT(1),ISXST), (IT(21),ISH), (IT(41),RHO)
     1            ,(IT(61),NTAPE1), (IT(62),NTAPE2), (IT(63),NTAPE3),
     2             (IT(64),NTAPE4), (IT(65),NTAPE5), (IT(66),NTAPE6),
     3             (IT(67),NTAPE7), (IT(68),NTAPE8), (IT(69),NTAPE9),
     4             (IT(70),NSUR), (IT(71),NRIGID), (IT(72),BREF),
     5             (IT(73),NAERO), (IT(74),NFUS), (IT(75),NDENS),
     6             (IT(76),MODES), (IT(77),NPOINT), (IT(78),NPUNCH),
     7             (IT(79),MAXR), (IT(80),MAXQ), (IT(81),MAXS),
     8             (IT(82),NC), (IT(83),NSURFS), (IT(84),NO2),
     9             (IT(85),HR), (IT(106),S), (IT(127),VELCTY)
      EQUIVALENCE  (IT(147),NTAPE0), (IT(148),NRHO), (IT(149),NITRSP),
     1             (IT(150),NITRDP), (IT(151),EPSP), (IT(152),EPDP),
     2             (IT(153),PICON), (IT(154),NVEL), (IT(155),NCARDS),
     3             (IT(156),AITKEN), (IT(157),NGO), (F,G), (F(601,1),HT)
     4,              (U,LOW), (U(51,1),LHIGH), (IT(158),FK),(IT(159),SQR
     5CON)           , (IT(160),NSIZES), (IT(180),DM), (DMBAR,BARMRR)
     6               , (IT(216),NDELM),(IT(217),AITKED),(IT(218),KPART)
    1 FORMAT   (18I4)
    2 FORMAT   (6E12.8)
    3 FORMAT   (1H 16X, 41H FLUTTER ANALYSIS BY A COLLOCATION METHOD
     1          42H USING AERODYNAMIC INFLUENCE COEFFICIENTS ///10H NSUR =
     2          1I2, 10H  NAERO =  1I4, 11H  NRIGID = 1I2, 9H  NFUS =
     3          1I2, 10H  NDENS =  1I4, 14H  MODES OUT =  1I2
     4          , 9H NDELM = 1I2,10H NPUNCH =  1I2    )
    4 FORMAT (1H0 22X, 11H B (REF) = 1E20.8, 5X, 4HK = 1E20.8 /1H0 25X,
     1         7HSURFACE 18X, 1HB 19X, 1HS 10X,20HEXTERNAL STORES SIZE //)
    5 FORMAT   (1H0 10X, 21H B RIGID COMPONENT =  1E18.8, 5X, 8H S RIGID
     1           13H COMPONENT =  1E18.8  )
    6 FORMAT   (1H   1I29,  2(  5X,  1E20.8),  1I12 )
   10 FORMAT   (1H1  48X, 12H MASS MATRIX    )
   11 FORMAT   (41H0 NUMBER OF CONTROL POINTS THIS MATRIX, ( 1I4,
     1           48H) AND TOTAL NUMBER OF CONTROL POINTS EXPECTED, (  1I4,
     2           37H) DO NOT AGREE, PROGRAM CONTINUED....   )
   12 FORMAT   (1H1 42X,24H RIGID BODY MODAL MATRIX   )
   13 FORMAT   (1H  38X,  8HSURFACE  1I2,  1H,  1I6, 15H CONTROL POINTS)
   14 FORMAT   (1H1 43X, 20H FLEXIBILITY MATRIX    )
   15 FORMAT   (1H1  46X, 18H WEIGHTING MATRIX   )
   16 FORMAT   (1H1 28X, 29H RIGID COMPONENT AERO MATRIX, 1I9, 8H CONTROL
     1           7H POINTS     )
   17 FORMAT   (1H1    33X,  20H AERODYNAMIC MATRIX  8X, 10H 1./K R =
     1             1F20.8   )
   18 FORMAT   (1H0 30X, 23H RIGID COMPONENT MODES,  1I10,
     1          17H CONTROL POINTS.     )
   19 FORMAT   (1H0 30X, 29H RIGID COMPONENT MASS MATRIX    )
   20 FORMAT   (47H' ERROR IN INVERSE ROUTINE, PROGRAM TERMINATED      )
   21 FORMAT   ( 1H    38X, 8HSURFACE  1I2,  1H, 6X, 13H NO WEIGHTING
     1            7H MATRIX  )
   22 FORMAT   ( 3E12.8, 2I4 )
   23 FORMAT   ( 8H0EPSP =    1F16.8, 1X, 8H EPDP =   1E16.8, 1X,
     1           10H AITKEN =  1E16.8, 1X, 10H NITRSP = 1I4, 1X,
     2           10H NITRDP =     1I4   )
   24 FORMAT   (1H0 40X, 20H GENERALIZED MASSES  ////     (1H   30X,
     1             5H MASS  1I4,  3H =    1F16.8 )  )
```

```
   25 FORMAT   ( 1H0  34X, 23H RIGID COMPONENT MODES  ///     )
   26 FORMAT   ( 214, 62X, 1A6, 1I4 )
   27 FORMAT   ( 1H0  35X, 23H PUNCHED CARDS NUMBERS  1A6, 1I4, 6H THRU
     1              1A6, 1I4   )
   28 FORMAT (1H0  20X,  3H K= 1E16.8     )
      DATA Q000CT/0202030440005/
      BCDZ =Q000CT
C     NTAPE0 = PUNCH OUTPUT TAPE
C     NTAPE2 = INPUT TAPE
C     NTAPE3 = OUTPUT PRINT TAPE
C     NTAPE4 =   /
C     NTAPE5 =   /    ARE UTILITY TAPES
C     NTAPE6 =  /
C     NTAPE7 = /
C     NTAPE8 =/
C     NTAPE9=/
      IF  ( NGO-98765 )      99,97,99
   97 IF  ( NAERO )          98,320,98
   98 NAERO = NAERO-1
      IF  ( NAERO )          99,99,170
   99 NTAPE0 = 8
      NTAPE2=5
      NTAPE3=6
      NTAPE4 = 9
      NTAPE5 = 3
      NTAPE6 = 4
      NTAPE7=11
      NTAPE8=1
      NTAPE9 =10
      MAXR = 50
      MAXQ = 6
      NCON=0
      MAXS=50
  991 REWIND NTAPE4
      REWIND NTAPE5
      REWIND NTAPE6
      REWIND NTAPE9
      SQRCON =  SQRT( 386.0) / (2.0*3.14159 )
      EPSP = .5E-06
      EPDP = .5E-07
      AITKEN = .9
      AITKED = .9
      PICON = .5921*2.0*3.14159
      NCARDS = 0
      NVEL  = 0
      NC=1
      NITRSP=40
      NITRDP=100
      RHO(1) = 0.0
C*************************************************************************
C READ IN TITLE, CONTROLS AND CONSTANTS AND PRINT.
  100 CALL  RDIN   (NTAPE2,NTAPE3,1)
      READ   (NTAPE2,1)NSUR, NAERO, NRIGID, NFUS, NDENS, MODES          ,
     1NDELM, NPUNCH, NCON
      READ   (NTAPE2,2)FK
      WRITE  (NTAPE3,3)NSUR,NAERO, NRIGID,NFUS, NDENS,MODES
     1          , NDELM, NPUNCH
      NR2=NRIGID*NC
      NSURFS=NSUR+NFUS
      IF   ( NCON )           102,104,102
  102 READ   (NTAPE2,22)EPSP,EPDP,AITKEN, NITRSP,NITRDP
```

```
      READ  (NTAPE2,22)AITKED
      WRITE (NTAPE3,23)EPSP, EPDP, AITKEN, NITRSP, NITRDP
  104 IF  ( NAERO )              105,103,105
  103 WRITE (NTAPE3,28)FK
      GOTO  111
  105 READ  (NTAPE2,1)(ISXST(I),ISW(I),I=1,NSUR)
      READ  (NTAPE2,2)BREF,  (VELCTY(I),I=1,NAERO)
      NC=2
      NR2=  NRIGID *NC
      READ  (NTAPE2,2)(BR(I),S(I),I=1,NSURFS)
      IF ( NDENS )             106,107,106
  106 READ  (NTAPE2,2)(RHO(I),I=1,NDENS)
  107 IF ( NFUS )              108,109,108
  108 WRITE (NTAPE3,5)BR(1), S(1)
  109 WRITE (NTAPE3,4)BREF   ,FK
      DO  110 I=1,NSUR
        J=I+NFUS
  110   WRITE (NTAPE3,6) I, BR(J), S(J) ,ISXST(I)
C*******************************************************************
C READ IN FUSE   OF MASS CHARACTERISTICS
  111 IF  ( NRIGID )           112,117,112
  112 IF  ( NDELM  )           115,113,115
  113 DO  114 I=1,NRIGID
        DO   114 J=1,NRIGID
  114     DM(I,J)=0.0
      GOTO  117
  115 DO  116  I=1,NRIGID
  116   READ  (NTAPE2,2)(DM(J,I),J=1,NRIGID)
      WRITE (NTAPE3,19)
      CALL MPRINT (DM,NRIGID,NRIGID,MAXQ,NTAPE3)
C*******************************************************************
C READ MASS MATRIX FOR EACH SURFACE, STORE SYSTEM MASS MATRIX ON NTAPE4
  117 K1=0
        WRITE (NTAPE3,10)
      DO  120 I=1,NSUR
        READ  (NTAPE2,1)NSIZE
        READ  (NTAPE2,1)(LOW(J),LHIGH(J),J=1,NSIZE)
        DO  119  J=1,NSIZE
          DO  118  K=1,NSIZE
  118       A(K,J)=0.0
      N1=LOW(J)
      N2=LHIGH(J)
  119     READ  (NTAPE2,2)(A(N,J),N=N1,N2)
      WRITE (NTAPE4)NSIZE, NSIZE,  ((A(N,J),N=1,NSIZE),J=1,NSIZE)
        WRITE (NTAPE3,11)I, NSIZE
        CALL MPRINT (A,NSIZE,NSIZE,MAXR,NTAPE3)
      NSIZES(I)=NSIZE
  120   K1=K1+NSIZE
C*******************************************************************
C NPOINT = TOTAL NUMBER OF CONTROL POINTS ON ALL SURFACES.
      NPOINT=K1
C READ IN RIGID BODY MODAL MATRIX FOR EACH SYSTEM.
      IF ( NFUS )              121,123,121
  121 READ  (NTAPE2,1)NSIZE
      DO  122  J=1,NRIGID
  122   READ  (NTAPE2,2)(HR(I,J),I=1,NSIZE)
      WRITE (NTAPE3,18)NSIZE
      CALL MPRINT (HR,NSIZE,NRIGID,MAXS,NTAPE3)
      WRITE (NTAPE6)NSIZE, NRIGID, ((HR(I,J),I=1,NSIZE),J=1,NRIGID)
  123 IF ( NRIGID )            124,128,124
  124 K1=1
```

```
      WRITE (NTAPE3,12)
      DO  126  ISUR=1,NSUR
        READ  (NTAPE2,1)NSIZE
        K2=K1+NSIZE-1
        DO  125  J=1,NRIGID
  125     READ  (NTAPE2,2)(HR(I,J),I=K1,K2)
        WRITE (NTAPE3,13)ISUR,NSIZE
          CALL MPRINT  (HR(K1,1),NSIZE,NRIGID,MAXS,NTAPE3)
  126     K1=K1+NSIZE
      WRITE (NTAPE5)NPOINT, NRIGID, ((HR(N,J),N=1,NPOINT), J=1,
     1                       NRIGID  )
      IF  ( K1-NPOINT-1)     127,128,127
  127 WRITE (NTAPE3,11)K1, NPOINT
C**********************************************************************
C READ IN FLEXIBILITY MATRIX FOR EACH SURFACE,      STORE ON NTAPE5
  128 K1=0
        WRITE (NTAPE3,14)
      DO  153  I=1,NSUR
      READ  (NTAPE2,1)NSIZE, J, IFORM, IROWS
      CALL MREAD (A,NSIZE,NSIZE,IFORM,IROWS,0,1,F,MAXR,NTAPE2,NTAPE3)
        WRITE (NTAPE3,13)I,NSIZE
      CALL MPRINT (A,NSIZE,NSIZE,MAXR,NTAPE3)
      WRITE (NTAPE5)NSIZE, NSIZE, ((A(J,K),J=1,NSIZE),K=1,NSIZE)
  153 K1=K1+NSIZE
      IF ( K1-NPOINT )        158,164,158
  158 WRITE (NTAPE3,11)K1, NPOINT
  164 IF ( NAERO )            166,165,166
  165 NRHO=1
      DO  500  I=1,NRIGID
        DO  500  J=1,NR?
  500     DQ(I,J)=0.0
      WRITE (NTAPE5)((DM(I,J),I=1,NRIGID),J=1,NRIGID)
       NCX=NC
      GOTO  215
C**********************************************************************
C READ IN WEIGHTING MATRIX FOR EACH SURFACE.....STORE ON NTAPE6,
  166  WRITE (NTAPE3,15)
      DO  178  I=1,NSUR
       IF  ( ISW(I) )        167,177,167
  167  N1=ISYST(I)
       IF  ( N1 )            168,172,168
  168  DO  170  J=1,N1
         DO  169  L=1,MAXR
           A(J,L)=0.0
  169      A(L,J)=0.0
  170    A(J,J)=1.0
       READ  (NTAPE2,1)NXST, J, IFORM, IROW
       IF  ( NXST )          171,172,171
  171  CALL MREAD (A,NXST,NXST,IFORM,IROW,0,1,F,MAXR,NTAPE2,NTAPE3)
  172  K=N1+1
       READ  (NTAPE2,1)NSIZE, NPART, IFORM, IROW
       IF  ( IFORM )         174,173,174
  173  CALL MREAD  (A(K,K),NSIZE,NSIZE,0,0,0,1,F,MAXR,NTAPE2,NTAPE3)
       K=K+NSIZE
       GOTO  176
  174  DO  175  J=1,NPART
         READ  (NTAPE2,1)NSIZE
      N=K+NC
      DO  1741  L=K,N
        DO  1741  M=1,K
 1741     A(M,L)=0.0
```

```
      CALL MREAD (A(K,K),NSIZE,NSIZE,1,IROW,0,0,F,MAXR,NTAPE2,NTAPE3)
  175   K=K+NSIZE
  176   K=K-1
C*********************************************************************
      WRITE (NTAPE3,15)I, K
      CALL MPRINT (A,K,K,MAXR,NTAPE3)
      WRITE (NTAPE6)K, K, ((A(J,L),J=1,K),L=1,K)
      GOTO  178
  177 WRITE (NTAPE3,21)I
  178   CONTINUE
  179 REWIND NTAPE6
      REWIND NTAPE7
      NCX=NC
      NVEL = NVEL+1
C*********************************************************************
C IF SERIES OF DENSITIES FOR EACH V/R OMEGA, READ IN THAT SERIES.
      IF  ( NDENS )         180,181,180
  180 NRHO=NDENS
      GOTO  182
  181 READ  (NTAPE2,1)NRHO
      READ  (NTAPE2,2)(RHO(I),I=1,NRHO)
C*********************************************************************
C READ IN COMPLEX AERODYNAMIC MATRIX FOR EACH SURFACE
  182 DO  210  I=1,NSURFS
      K=1
      K2=1
      IF  ( NFUS )           183,184,183
  183 IF  ( I-1 )            184,188,184
  184 L=I-NFUS
      IF  ( ISXST(L) )       185,188,185
  185   K=ISXST(L)+1
        K2 = K+ISXST(L)
        DO  186  J=1,K
          DO  186  L=1,K2
  186       A(J,L)=0.0
        READ  (NTAPE2,1)NXST, J, IFORM, IROW
        IF  ( NXST )         187,188,187
  187   N=  NXST *NC
      CALL MREAD (A,NXST,N,IFORM,IROW,0,1,U,MAXR,NTAPE2,NTAPE3)
  188   READ  (NTAPE2,2)VELC
        READ  (NTAPE2,1)NSIZE, NPART, IFORM, IROW
        IF  ( IFORM )        190,189,190
  189   N=  NSIZE      *NC
      CALL MREAD (A(K,K2),NSIZE,N ,IFORM,IROW,0,1,U,MAXR,NTAPE2,NTAPE3)
      NSIZE=NSIZE+K-1
        GOTO  193
  190   DO  192  J=1,NPART
          READ  (NTAPE2,1)NSIZE
      N=K2+NC
      DO  191  M=K2,N
        DO  191  L=1,K
  191     A(L,M)=0.0
          N=  NSIZE      *NC
      CALL MREAD (A(K,K2),NSIZE,N,1,IROW,0,0,U,MAXR,NTAPE2,NTAPE3)
          K=K+NSIZE
  192     K2=K2+N
      NSIZE=K-1
          N=K2-1
  193 IF  ( I-1 )            199,194,199
  194 IF  ( NFUS )           195,197,195
  195   WRITE (NTAPE3,15)NSIZE
```

```
      CALL  MPRINT (A,NSIZE,N,MAXR,NTAPE3)
C*****************************************************************************
C   COMPUTE (DQ) = (HR)T * (CH) * (HR)
      K1=1
      K2=1
      M=NC-1
      READ  (NTAPE6)NSIZE, L, ((U(L,J),L=1,NSIZE),J=1,NRIGID)
      CALL  MMULTD  (A,M,U,0,F,NSIZE,NSIZE,NRIGID,MAXR,MAXR,MAXR)
      DO  196  L=1,NSIZE
        DO  196  J=1,NRIGID
  196     A(J,L)=U(L,J)
      CALL MMULTD (A,0,F,M,DQ,NRIGID,NSIZE,NRIGID,MAXR,MAXQ,MAXQ)
      GOTO  210
  197 DO  198  L=1,NRIGID
        DO  198  J=1,NR2
  198     DQ(L,J)=0.0
  199 I=I-NFUS
      IF  ( ISW(L) )          202,200,202
  200 WRITE (NTAPE7)NSIZE, N,     ((A(J,M),J=1,NSIZE),M=1,N)
      GOTO  208
  202 READ  (NTAPE6)L, L, ((F(J,M),J=1,L),M=1,L)
      IF ( NSIZE-L )          204,206,204
  204 WRITE (NTAPE3,11)NSIZE,L
  206 CALL  MMULTD (F,0,A,NC-1,U,L,L,L,MAXR,MAXR,MAXR)
      WRITE (NTAPE7)NSIZE, N, ((U(J,M),J=1,NSIZE),M=1,N)
  208 WRITE (NTAPE3,17)VELC
      L=I-NFUS
      WRITE (NTAPE3,13)L, NSIZE
      CALL MPRINT  (A,NSIZE,N,MAXR,NTAPE3)
  210 CONTINUE
C*****************************************************************************
C CARRY ON FROM HERE TO END ONCE FOR EACH DENSITY.
  215 DO  300  IRHO=1,NRHO
      K=NC*NPOINT
      DO  216  I=1,NPOINT
        DO  216  J=1,K
  216     U(I,J)=0.0
      REWIND  NTAPE4
      REWIND NTAPE5
      REWIND NTAPE7
        CON=RHO(IRHO)*BR(1)**2*S(1)  *32.174
      IF  ( NRIGID)     218,222,218
  218   DO  220  I=1,NRIGID
          DO  220  J=1,NR2,NCX
            DMBAR(I,J+1)=CON*DQ(I,J+1)
            K=J/NC+NC-1
  220       DMBAR(I,J)=DM(I,K)+CON*DQ(I,J)
C*****************************************************************************
C READ ENTIRE HR MATRIX.
      READ  (NTAPE5)I,I, ((HR(I,J),I=1,NPOINT),J=1,NRIGID)
  222  K1=0
      DO  246  ISUR=1,NSUR
         K=ISUR+NFUS
         CON=RHO(IRHO)*BR(K)**2*S(K) *32.174
C*****************************************************************************
C                READ  (M) I
         READ  (NTAPE4)NSIZE, NSIZE, ((F(I,J),I=1,NSIZE),I=1,NSIZE)
      IF ( NAERO )            221,221,223
  221 DO  226  I=1,NSIZE
        DO  226  J=1,NSIZE
  226     A(I,J)=F(I,J)
```

```
      I=NSIZE
      GOTO  231
C*****************************************************************************
C              READ  (H)*(CH) I
  243     READ  (NTAPE7)NSIZE, L,  ((A(I,J),I=1,NSIZE),J=1,L)
C*****************************************************************************
C         COMPUTE (M BAR) =((H)+RHO*RR**2*S*(H)*(CH) ) I
      DO  228  I=1,NSIZE
        DO  228  J=1,L,NCX
          K=J/2+1
          A(I,J)=CON*A(I,J)+F(I,K)
  228     A(I,J+1)=CON*A(I,J+1)
C*****************************************************************************
C READ FLEXIBILITY MATRIX FOR SURFACE
  231 READ  (NTAPE5)NSIZE, NSIZE, ((F(I,J),I=1,NSIZE),J=1,NSIZE)
      K=NC*K1+1
      CALL MMULTD (F,0,A,NC-1,U(K1+1,K),NSIZE,NSIZE,NSIZE,MAXR,MAXR,
     1             MAXR)
      IF  ( NRIGID )          232,246,232
C*****************************************************************************
C FIND ( LITTLE M BAR ) = (DELTA M ) + (H R)TRANSPOSED * (M BAR) * (H R)
  232   DO  240  I=1,NSIZE
          K=K1+I
        DO  240  J=1,NRIGID
  240     G(J,I)=HR(K,J)
      K=NC*K1+1
      CALL MMULTD (G,0,A,NC-1,HRT(1,K),NRIGID,NSIZE,NSIZE,MAXQ,MAXR,
     1             MAXQ)
      CALL MMULTD  (HRT(1,K),NC-1,HR(K1+1,1),0,A,NRIGID,NSIZE,NRIGID,
     1                MAXQ,MAXS,MAXR)
      DO  244  I=1,NRIGID
        DO  244  J=1,NR2
  244     BARMRR(I,J)=BARMRR(I,J)+A(I,J)
  246 K1=K1+NSIZE
      L=NC*NPOINT
      IF  ( NRIGID )          247,268,247
  247 GOTO  (252,248),NC
  248 DO  250  I=1,NRIGID
        DO  250  J=1,NR2,NCX
          K=J/2+1
          G(I,K)=BARMRR(I,J+1)
  250     BARMRR(I,K)=BARMRR(I,J)
C*****************************************************************************
C THEN (LITTLE M BAR)INVERSE AND FINAL U MATRIX STORED ON NTAPE9.
  252 CALL MNVRSX (BARMRR,G,A(1,1), A(1,MAXQ),NRIGID,IR,NC-1)
      IF  ( IR )              310,254,310
  254 GOTO  (260,256),NCX
  256 DO  258  I=1,NRIGID
        DO  258  J=1,NR2,NCX
          K=NR2-J
          M=K/2+1
          BARMRR(I,K)=BARMRR(I,M)
  258     BARMRR(I,K+1)=G(I,M)
  260 CALL MMULTD  (BARMRR,NC-1,HRT,NC-1, F,NRIGID,NRIGID, POINT,
     1                MAXQ,MAXQ,MAXR)
      CALL MMULTD  (F,NC-1,U,NC-1,HRT,NRIGID,NPOINT,NPOINT,MAXR,MAXR,
     1                MAXQ)
      CALL MMULTD  (HR,1,F,NC-1,A,NPOINT,NRIGID,NPOINT,MAXS,MAXR,MAXR)
      DO  264  I=1,NPOINT
        DO  262  J=1,L
          F(I,J)=U(I,J)
```

```
262       A(I,J)=-A(I,J)
        K=I*NC-NC+1
264     A(I,K)=1.0+A(I,K)
      CALL MMULTD (A,NC-1,F,NC-1,U,NPOINT,NPOINT,NPOINT,MAXR,MAXR,MAXR)
      IF  ( NAERO )           268,266,268
266   READ  (NTAPE5)A(1,1)
      WRITE (NTAPE5)((HRT(I,J),I=1,NRIGID),J=1,NPOINT)
268   WRITE (NTAPE9)NPOINT, L, ((U(I,J),I=1,NPOINT),J=1,L)
300   CONTINUE
      REWIND  NTAPE5
      REWIND  NTAPE6
      REWIND  NTAPE7
      REWIND  NTAPE8
      REWIND  NTAPE9
      KPART = 2
      RETURN
310   WRITE (NTAPE3,20)
      GOTO  991
320   REWIND  NTAPE4
      REWIND  NTAPE5
      IF  ( NRIGID )          330,332,330
330   READ  (NTAPE5)I,I, A(1,1)
332   K1=1
      DO  334  I=1,NPOINT
        DO  334  J=1,NPOINT
334       F(I,J)=0.0
      DO  336  ISUR=1,NSUR
        K2=K1+NSIZES(ISUR)-1
        READ  (NTAPE5)I,I, A(1,1)
        READ  (NTAPE4)NSIZE, NSIZE, ((F(I,J),I=K1,K2),J=K1 K2)
336     K1=K1+NSIZE
      READ  (NTAPE4)NSIZE, MODE, ((A(I,J),I=1,NSIZE),J=1,MODE)
      M=MODE/NC
      CALL MMULTD  (F,0,A,NC-1,U,NPOINT,NPOINT,M,MAXR,MAXR,MAXR)
      IF  ( NDELM )           337,348,337
337   IF  ( NRIGID )          338,348,338
338   READ  (NTAPE5)((DM(I,J),I=1,NRIGID),J=1,NRIGID)
      READ  (NTAPE5)((G(I,J),I=1,NRIGID),J=1,NPOINT)
      DO  339  I=1 NRIGID
        DO  339  J=1,NPOINT
339       G(I,J)=-G(I,J)
      CALL MMULTD  (G,0,A,0,HRT,NRIGID,NPOINT,M,MAXQ,MAXR,MAXQ)
      READ (NTAPE4)  (HR(I,1),I=1,M)
      DO  340  I=1,M
        DO  340  J=1,NRIGID
340       G(J,I) = HRT(J,I)/HR(I,1)
      WRITE (NTAPE3,25)
      CALL MPRINT (G,NRIGID,M,MAXQ,NTAPE3)
      IF  ( NPUNCH )            343,345,343
343   WRITE (NTAPE0,26)NRIGID, M, BCDZ, NCARDS
      NCARD = NCARDS+1
      CALL MPUNCH  ( G,NRIGID,M,0,1,1,BCDZ,MAXQ,NTAPE0,NCARD)
      WRITE (NTAPE3,27)BCDZ, NCARDS, BCDZ, NCARD
      NCARDS=NCARD+1
345   CONTINUE
      CALL MMULTD  (DM,0,G,0,HRT,NRIGID,NRIGID,M,MAXQ,MAXQ,MAXQ)
      DO  346  I=1,NRIGID
        DO  346  J=1,MODE
346       HR(J,I) = G(I,J)
348   DO  350  I=1,NPOINT
        DO  350  J=1,MODE
```

```
350       F(J,I)=A(I,J)
      CALL MMULTD  (F,0,U,0,A,M,NPOINT,M,MAXR,MAXR,MAXR)
      IF  ( NDELM )            352,358,352
  352 IF  ( NRIGID )           354,358,354
  354 CALL MMULTD  (HR,0,HRT,0,F,M,NRIGID,M,MAXR,MAXQ,MAXR)
      DO  356  I=1,MODE
        DO  356  J=1,M
  356     A(J,I) = A(J,I) + F(J,I)
  358 WRITE (NTAPE3,24) ( I, A(L,I),I=1,M)
      GOTO  991
      END
       FORTRAN LSTOU.DECK
C MREAD
C   MREAD
C MATRIX READ SUBROUTINE
C     CALL MREAD   (A,M,N,IFORM,IROW,ITRA,IORG,T,MD,NTAPE2,NTAPE3      )
C
C          A = MATRIX TO READ IN         ITRA = 0, TRA CARD AFTER MATRIX
C          M = NUMBER OF ROWS                 =+1, TRA CARD AFTER EACH ROW
C          N = NUMBER OF COLUMNS                        (OR COLUMN )
C          IFORM = -1, FORMAT(12A6)      IORG = ORIGIN OF FIRST C.B. CARD
C                =  0, COLUMN BINARY     T =MDXN TEMPORARY CELLS
C                = +1, FORMAT(6E12.8)    MD= DIMENSIONED NUMBER OF ROWS
C          IROW = .0, MATRIX BY COLUMNS                          IN A
C               = +1, MATRIX BY ROWS     NTAPE2 = INPUT TAPE
C                                        NTAPE3 = OUTPUT TAPE
      SUBROUTINE MREAD  (A,M,N,IFORM,IROW,ITRA,IORG,T,MD,NTAPE2,NTAPE3 )
      DIMENSION    A(1), T(1)
    1 FORMAT   (6E12.8)
    2 FORMAT   (12A6)
    3 FORMAT   ( // 26H THATS ALL YOUR DATA.            )
    4 FORMAT (4E16.8)
      MN=MD*N
      DO  5  I=1,MN
        T(I)=0.0
    5   A(I)=0.0
      IF ( IFORM )  39,15,6
    6 IF  ( IROW )   8,7,8
    7 K3=1
      K4=N
      K5=MD
      K6=M-1
       K2=1
      GOTO 9
    8 K2=MN
      K3=MD
      K4=M
      K5=1
      K6=0
    9 DO  11  I=1,K4
        K1=I*K5-K5+1
        IF  (K6)   10,111,10
   10   K2=K1+K6
  111 IF (IFORM - 1)  38,110,109
  109 READ (NTAPE2,4) (A(L) ,L = K1,K2,K3)
      GO TO 11
  110   READ  (NTAPE2,1)(A(L),L=K1,K2,K3)
   11 CONTINUE
        GOTO 36
   15 K1=N
      K2=M
```

```
      K3=1
      IF ( IORG-1 )  16,17,17
   16 K3=2
   17 IF  ( IROW )  18,19,18
   18 K2=N
      K1=M
   19 IF ( ITRA )    22,21,22
   21 K1=1
   22 K=0
      DO  23  I=1,K1
        K4=K+K3
        K5=1
      CALL  BINRD  (T(K4), K5, L, NTAPE2, NTAPE3 )
        GOTO  (23,38,23,23),L
   23 K=K+K2
      IF  ( IROW )  28,24,28
   24 L=0
      IF  ( IORG-1)  26,26,25
   25 L=IORG-1
   26 DO  27  I=1,N
        J=I*MD-MD
        DO  27  K=1,M
          J=J+1
      L=L+1
   27 A(J)=T(L)
      GOTO  36
   28 L=0
      IF  (IORG-1)  30,30,29
   29 L=IORG-1
   30 DO  31  K=1,N
        J= K*MD-MD
        DO  31  I=K,MN,N
          J=J+1
          K1=L+I
   31     A(J)=T(K1)
   36 RETURN
   38 WRITE (NTAPE3,3)
      STOP
   39 READ  (NTAPE2,2)(A(I),I=1,M)
      GOTO  36
      END
$       FORTRAN LSIOU.DECK
C BINRD
      SUBROUTINE BINRD (T,K,L,NTAPE1,NTAPE2)
      DIMENSION T(1)
      RETURN
      END
$       FORTRAN LSIOU.DECK
C RDLN
      SUBROUTINE  RDLN  (NTAPE2, NTAPE3, I )
    1 FORMAT(80H
     1                                          )
    2 FORMAT(1H1)
    3 FORMAT  ( 1H0 )
      READ  (NTAPE2,1)
      GOTO   (4,5),I
    4 WRITE  (NTAPE3,2)
      GOTO   6
    5 WRITE  (NTAPE3,3)
    6 WRITE  (NTAPE3,1)
      RETURN
```

```
      END
      FORTRAN LSTOU.DECK
C MMULTD
      SUBROUTINE MMULTD  (A,N1,B,N2,C,M,N,K,MA,MB,MC)
      DIMENSION  A(1), B(1), C(1)
      IC=1
      IA=MC*K
      IB=MA*N
      ID=MA
      IH=MC
      IJ=MC
      IF ( N1 )  4,3,4
    3 IF ( N2 )  7,8,7
    4 IB=2*IB
      ID=2*ID
      IF ( N2 )  5,6,5
    5 IC=2
      GOTO  7
    6 IH=2*IH
      IC=3
    7 IA=2*IA
      IJ=2*IJ
    8 DO  18  I=1,M
      INC=0
      DO  11  J=I,IA,IH
        C(J)=0.0
        IN=INC
        DO  10  L=I,IB,ID
          IN=IN+1
   10     C(J)=C(J)+A(L)*B(IN)
   11   INC=INC+MB
      INC=0
      GOTO  (18,12,15),IC
   12 DO  14  J=I,IA,IJ
        IF=I+MA
        IE=J+MC
        IN=INC
        DO  13  L=IF,IB,ID
          IN=IN+1
          IG=IN+MB
          C(IE)=C(IE)+A(L)*B(IN)
   13     C(J)=C(J)-A(L)*B(IG)
   14   INC=INC+MB *2
        GOTO  18
   15 IE=I+MC
      IF=I+MA
      DO  17  J=IE,IA,IJ
        IN=INC
        C(J)=0.0
        DO  16  L=IF,IB,ID
          IN=IN+1
   16     C(J)=C(J)+A(L)*B(IN)
   17   INC=INC+MB
   18 CONTINUE
      RETURN
      END
       FORTRAN LSTOU.DECK
C MNVRSX
      SUBROUTINE  MNVRSX (AR,AI,B,C,KSZ,IGOOFD,NOP)
      DIMENSION AR(6,6), AI(6,6), B(6,6), C(6,6)
      IGOOFD=0
```

```
      IF ( NOP )      100,101,102
  101 CALL INVERS  (AR,KSZ,IGOOFD)
      GO TO 20
  102 CONTINUE
      DO 1 K=1,KSZ
      DO 1 L=1,KSZ
    1 B(K,L)=AR(K,L)
      NO=0
      CALL INVERS(B,KSZ,NO)
      IF (NO) 2,3,2
C  REAL MATRIX NOT SINGULAR
C  MULT B*AI   STO. C
    3 DO 4 K=1,KSZ
      DO 4 L=1,KSZ
      C(K,L)=0.0
      DO 4 I=1,KSZ
    4 C(K,L)=C(K,L)+B(K,I)*AI(I,L)
C  MULT. AI*C + AR   STO. B
      DO 5 K=1,KSZ
      DO 5 L=1,KSZ
      B(K,L)=AR(K,L)
      DO 5 I=1,KSZ
    5 B(K,L)=B(K,L)+AI(K,I)*C(I,L)
      NO=0
      CALL INVERS(B,KSZ,NO)
      IF (NO) 2,7,2
C  SECOND MATRIX  NOT SINGULAR
C  MULT. -C*B  STO. AI      ALSO SET AR=B
    7 DO 8 K=1,KSZ
      DO 8 L=1,KSZ
      AI(K,L)=0.0
      AR(K,L)=B(K,L)
      DO 8 I=1,KSZ
    8 AI(K,L)=AI(K,L)-C(K,I)*B(I,L)
      GO TO 20
C  REAL MATRIX OR SECOND MATRIX SINGULAR    TRY IMAG. ROUT
    2 DO 9 K=1,KSZ
      DO 9 L=1,KSZ
    9 B(K,L)=AI(K,L)
      NO=0
      CALL INVERS(B,KSZ,NO)
      IF (NO) 10,11,10
C  IMAG. NOT SINGULAR
C  MULT. B*AR   STO. C
   11 DO 12 K=1,KSZ
      DO 12 L=1,KSZ
      C(K,L)=0.0
      DO 12 I=1,KSZ
   12 C(K,L)=C(K,L)+B(K,I)*AR(I,L)
C  MULT. AR*C+AI   STO B
      DO 13 K=1,KSZ
      DO 13 L=1,KSZ
      B(K,L)=AI(K,L)
      DO 13 I=1,KSZ
   13 B(K,L)=B(K,L)+AR(K,I)*C(I,L)
      NO=0
      CALL INVERS(B,KSZ,NO)
      IF (NO) 10,15,10
C  THIRD MATRIX NOT SINGULAR
C  MULT -C*B   STO AR    ALSO SET AI=-B
   15 DO 16 K=1,KSZ
```

```
      DO 16 L=1,KSZ
      AR(K,L)=0
      AI(K,L)=-B(K,L)
      DO 16 I=1,KSZ
16    AR(K,L)=AR(K,L)+C(K,I)*B(I,L)
      GO TO 20
10    I GOOF D=1
20    RETURN
      END
$      FORTRAN LSTOU,DECK
 INVERS
      SUBROUTINE INVERS (A,N,I GOOF D)
      DIMENSION A(6,6), L(6), M(6)
      CALL OVERFL(K000FX)
       GO TO(500,500),K000FX
500   CALL OVERFL(K000FX)
       GO TO(501,501),K000FX
501   CALL DVCHK (K000FX)
       GO TO(502,502),K000FX
502   IGOOFD=0
C     SEARCH FOR LARGEST ELEMENT
      DO 80 K=1,N
      L(K)=K
      M(K)=K
      BIGA=A(K,K)
      DO 20 I=K,N
      DO 20 J=K,N
      IF(ABS(BIGA)-ABS(A(I,J)))10,20,20
10    BIGA=A(I,J)
      L(K)=I
      M(K)=J
20    CONTINUE
C     INTERCHANGE ROWS
      JROW=L(K)
      IF(L(K)-K)35,35,25
25    DO 30 I=1,N
      HOLD=-A(K,I)
      A(K,I)=A(JROW,I)
30    A(JROW,I)=HOLD
C     INTERCHANGE COLUMNS
35    ICOL=M(K)
      IF(M(K)-K)45,45,37
37    DO 40 J=1,N
      HOLD=-A(J,K)
      A(J,K)=A(J,ICOL)
40    A(J,ICOL)=HOLD
C     DIVIDE COLUMN BY MINUS PIVOT
45    DO 55 IC=1,N
46    IF(IC-K)50,55,50
50    A(IC,K)=A(IC,K)/(-A(K,K))
55    CONTINUE
C     REDUCE MATRIX
      DO 65 I=1,N
      DO 65 J=1,N
56    IF(I-K)57,65,57
57    IF(J-K)60,65,60
60    A(I,J)=A(I,K)*A(K,J)+A(I,J)
65    CONTINUE
C     DIVIDE ROW BY PIVOT
      DO 75 JR=1,N
68    IF(JR-K)70,75,70
```

```
 71    A(K,JR)=A(K,JR)/A(K,K)
 75    CONTINUE
C      CONTINUED PRODUCT OF PIVOTS
C      REPLACE PIVOT BY RECIPROCAL
       A(K,K)=1.0/A(K,K)
C      CONTINUE COMPLETE OPERATION
 80    CONTINUE
       CALL DVCHK (K000FX)
        GO TO(510,503),K000FX
 503   CALL OVERFL(K000FX)
        GO TO(510,504),K000FX
 504   CALL OVERFL(K000FX)
        GO TO(510,505),K000FX
C      FINAL ROW AND COLUMN INTERCHANGE
 505   K=N
 100   K=(K-1)
       IF(K)150,150,103
 103   I=L(K)
       IF(I-K)120,120,105
 105   DO 110 J=1,N
       HOLD=A(J,K)
       A(J,K)=-A(J,I)
 110   A(J,I)=HOLD
 120   J=M(K)
       IF(J-K)100,100,125
 125   DO 130 I=1,N
       HOLD=A(K,I)
       A(K,I)=-A(J,I)
 130   A(J,I)=HOLD
       GO TO 100
 150   RETURN
 510   IGOOFD=1
       GO TO 150
       END
$       LINK    PAR2TT,PAR1TT
$       FORTRAN LSTOU,DECK
C PART2
       SUBROUTINE PART2
C PART.....2 VIBRATION AND FLUTTR ANALYSIS BY A COLLOCATION METHOD.
       DIMENSION IT(219),VELCTY(20),NSIZES(20),DM(6,6),RHO(20)
       COMMON/ I /          U(49,196),GUESS(49,2),H(49,50),EIG(50),
      1          TEMP(2734),             NAKSR(25), NAKDR(25), NITER(25),
      2          OMEGA(25), DAMP(25), VELC(25)
C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR TO
C   COMPENSATE FOR THE FACT THAT EQUIVALENCE DOES NOT REORDER COMMON---
       COMMON  IT
       EQUIVALENCE  (IT(1),ISXST), (IT(21),ISW), (IT(41),RHO),
      1             (IT(61),NTAPE1), (IT(62),NTAPE2), (IT(63),NTAPE3),
      2             (IT(64),NTAPE4), (IT(65),NTAPE5), (IT(66),NTAPE6),
      3             (IT(67),NTAPE7), (IT(68),NTAPE8), (IT(69),NTAPE9),
      4             (IT(70),NSUR), (IT(71),NRIGID), (IT(72),BREF),
      5             (IT(73),NAERO), (IT(74),NFUS), (IT(75),NDENS),
      6             (IT(76),MODES), (IT(77),NPOINT), (IT(78),NPUNCH),
      7             (IT(79),MAXR), (IT(80),MAXO), (IT(81),MAXS),
      8             (IT(82),NC), (IT(83),NSURFS), (IT(84),NP2),
      9             (IT(85),BR), (IT(106),S)
       EQUIVALENCE  (IT(127),VELCTY), (IT(147),NTAPE0), (IT(148),NRHO),
      1             (IT(149),NITRSP), (IT(150),NITRDP), (IT(151),EPSP),
      2             (IT(152),EPDP), (IT(153),PICON), (IT(154),NVEL),
      3             (IT(155),NCARDS), (IT(156),AITKEN), (IT(157),NGO),
      4             (IT(158),FK), (IT(159),SORCON), (IT(160),NSIZES),
```

```
     6              (IT(180),DM),(IT(217),AITKED), (IT(218),KPART)
 200 FORMAT  (1H1  48X,  12H OUTPUT DATA  /// 1H  11X,   6H FLUTTER
    1          52H ANALYSIS BY A COLLOCATION METHOD, USING AERODYNAMIC
    2          24H INFLUENCE COEFFICIENTS  // 1H0  14X, 11H DENSITY =
    3          1E20.8,  5X,  20H REDUCED VELOCITY =  1E20.8, // 1H0 33X,
    4          1I6,  32H RIGID BODY DEGREES OF FREEDOM   ///// )

 201 FORMAT  (1H1 20X, 5H MODE  7X, 12H OMEGA (CPS)  10X, 8H DAMPING
    1           4X, 17H VELOCITY (KNOTS)  ///   (   1H   2-X, 1I4,
    2           3F20.8)    )
 202 FORMAT  ( 2I4, 62X, 1A6, 1I4 )
 203 FORMAT  ( 1H0 33X, 23H PUNCHED CARDS NUMBERS  1A6, 1I4, 6H THRU
    1           1A6, 1I4   )
 204 FORMAT  (1H1  45X, 16H DYNAMIC MATRIX    )
 205 FORMAT  (31H FLEXIBLE MODE SHAPES, SURFACE     1I4   )
 206 FORMAT (5H MODE  1I6, 32H, GIVES AN IMAGINARY FREQUENCY.   )
     DATA Q000CT/02020304400057
     BCDZ =Q000CT
     NVEL=NVEL
     MAXP=49
     K=NC*NPOINT
     DO  290  I=1,NPOINT
       DO  290  J=1,K
 290     U(I,J)=0.0
     REWIND NTAPE9
     DO  314  IRHO=1,NRHO
       MODE=MODES
     READ  (NTAPE9)NSIZE, NSIZE2, ((U(I,J),I=1,NSIZE),
    1                      J=1,NSIZE2)
     IF  ( NPUNCH )           302,304,304
 302 WRITE (NTAPE3,204)
     CALL MPRINT (U,NPOINT,K,MAXP,NTAPE3)
 304 WRITE (NTAPE3,200)RHO(IRHO), VELCTY(NVEL), NRIGID
     DO  305  I=1,MODE
       NAKDR(I)=0
 305   NAKSR(I)=0
     CALL MITERS (U,GUESS ,0      ,NPOINT,MODE  ,MAXP  ,NC     ,EPSP  ,
    1              EPDP  ,NAKSR ,NAKDR ,NITRSP,NITRDP,AITKES,AITKED,
    2              IR    ,TEMP  ,H     ,EIG   ,NITER ,NTAPE9,NTAPE3)
     IF  ( NAERO )           3055,3054,3055
3054 WRITE (NTAPE4)  NPOINT, MODE, ((H(I,J),I=1,NPOINT),J=1,MODE)
     WRITE (NTAPE4)  (EIG(I),I=1,MODE),NPOINT,MODE,NPOINT,MODE,MODE
3055 CONTINUE
     MODES2 = NC*MODE
     DO  310  I=1,MODES2,NC
       K=I/NC +NC -1
     IF  (EIG(I))            3051,3052,3052
3051 WRITE (NTAPE3,206)K
     OMEGA(K)=0.0
     GOTO  3053
3052 OMEGA(K)= SQRCON / ( SQRT( FK*EIG(I)) )
3053 GOTO  (306,308),NC
 306   DAMP(K)=0.0
       VELC(K)=0.0
       GOTO  310
 308   DAMP(K)= EIG(I+1) / EIG(I)
       VELC(K)= PICON*OMEGA(K)*BREF* VELCTY(NVEL)
 310 CONTINUE
     WRITE (NTAPE3,201)(K, OMEGA(K), DAMP(K), VELC(K) ,
    1                   K=1,MODE )
     IF  ( NPUNCH )          312,314,312
```

```
  312 K1=1
      WRITE (NTAPE0,202)NSIZES(1), MODES2, BCDZ, NCARDS
      NCRDS = NCARDS+1
      CALL MPUNCH (OMEGA,MODES2,1,0,1,1,BCDZ,MAXP,NTAPE0,NCRDS)
      WRITE (NTAPE5,203)BCDZ, NCARDS, BCDZ, NCRDS
      NCARDS=NCRDS+1
      DO  313  ISUR =1,NSUR
      IF  ( ISUR-1 )             3121,3122,3121
 3121 WRITE (NTAPE0,202)NSIZES(ISUR), MODES2,BCDZ,NCARDS
      NCRDS=NCARDS+1
      GO TO  3123
 3122 NCRDS=NCARDS
 3123 CALL MPUNCH (H(K1,1),NSIZES(ISUR),MODSS2,0,1,1,BCDZ,MAXP,NTAPE0,
     1              NCRDS )
      WRITE (NTAPE5,205)ISUR
      WRITE (NTAPE5,203)BCDZ,NCARDS,BCDZ,NCRDS
      NCARDS=NCRDS+1
  313 K1=K1+NSIZES(ISUR)
  314 CONTINUE
      REWIND NTAPE0
      NGO = 98765
      KPART = 1
      RETURN
      END
$      FORTRAN LSTOU,DECK
C NORM7
      SUBROUTINE  NORM  (A,B,N,C,INDEX,MAXR,NC,NP)
      DIMENSION  A(1), B(1), C(1), T(2)
      IF  ( INDEX )                       100,200,200
  100 GO TO  (110,140),NP
  110 GO TO  (500,120),NC
  120 BIG = C(1)**2+C(2)**2
      GO TO  600
  140 CALL DNORM (A,B,N,C,MAXR,NC,T)
      GO TO 700
  200 INDEX=1
      NSTART=NP+1
      NSTOP=N*NP
      K=NP*MAXR
      IF  ( NSTART-NSTOP )                205,205,400
  205 GO TO  (210,300),NC
  210 BIG=ABS(A(1))
      DO  230   =NSTART,NSTOP,NP
      IF  ( BIG-ABS(A(I)) )               220,220,230
  220   INDEX=I
        BIG=ABS(A(I))
  230   CONTINUE
      GO TO  400
  300 BIG = A(1)**2 + A(K+1)**2
      DO  330   I=NSTART,NSTOP,NP
        J=K+I
      IF  ( BIG-(A(I)**2+A(J)**2) )                  320,320,330
  320   BIG = A(I)**2 + A(J)**2
        INDEX = I
  330   CONTINUE
  400 J=NP*NC
      I=INDEX+(NC-1)*K+(NP-1)
      C(J)=A(I)
      J=J-(NP-1)
      I=I-(NP-1)
      C(J)=A(I)
```

```
      I=INDEX+(NP 1)                                                    MTR40046
      C(NP)=A(I)                                                        MTR40047
      C(1)=A(INDEX)                                                     MTR40048
      GO TO  (410,140),NP                                               MTR40050
  410 GO TO  (500,600),NC                                               MTR40051
  500 DO  510  I=1,N                                                    MTR40053
  510   B(I)=A(I)/C(1)                                                  MTR40054
      GO TO  700                                                        MTR40055
  600 BIG = C(1)**2+C(2)**2                                             MTR40057
      DO  610  I=1,N                                                    MTR40058
        J=I+MAXR                                                        MTR40060
        T= A(I)*C(1)+A(J)*C(2)                                          MTR40061
        B(J) = ( A(J)*C(1)-A(I)*C(2) )  / BIG                           MTR40062
  610   B(I)= T/BIG                                                     MTR40063
  700 INDEX = INDEX/NP +NP-1                                            MTR40064
      RETURN                                                            MTR40065
      END                                                               MTR40067
$      FORTRAN LSIOU,DECK
C DNORMZ
      SUBROUTINE DNORM  (A,B,N,C,MAXR,NC,T)                             MTR40069
      DOUBLE PRECISION  A(1), B(1),C(1),T(1)                            MTR40071
  410 GO TO  (500,600),NC                                               MTR40072
  500 DO  510  I=1,N                                                    MTR40074
  510   B(I)=A(I)/C(1)                                                  MTR40075
      GO TO  700                                                        MTR40076
  600 BIG=C(1)*C(1) + C(2)*C(2)                                         MTR40078
      DO  610  I=1,N                                                    MTR40079
        J=I+MAXR                                                        MTR40080
        T= A(I)*C(1)+A(J)*C(2)                                          MTR40081
        B(J) = ( A(J)*C(1)-A(I)*C(2) )  / BIG                           MTR40082
  610   B(I)= T/BIG                                                     MTR40083
  700 RETURN                                                            MTR40084
      END                                                               MTR40086
$      FORTRAN LSIOU,DECK
C POHS
      SUBROUTINE   POH  (LMBDN,LMBD1,LMBD2,HN,HN1,H1,H2,N,NC)           MTR40089
      DIMENSION  LMBDN(1), LMBD1(1), LMBD2(1), HN(1), HN1(1), H1(1),    MTR40091
     1              H2(1), A(2)                                         MTR40092
      DOUBLE PRECISION  LMBDN, LMBD1, LMBD2, HN, HN1, H1, H2, A         MTR40093
      GO TO  (200,100),NC                                               MTR40095
          110  I=1,N                                                    MTR40097
        K=I+N                                                           MTR40099
        A(1) = LMBDN(1)*HN(I)-LMBDN(2)*HN(K)                            MTR40101
        A(2) = LMBDN(1)*HN(K)+LMBDN(2)*HN(I)                            MTR40102
        H1(I) = LMBD2(1)*HN1(I)-LMBD2(2)*HN1(K)-A(1)                    MTR40104
        H1(K) = LMBD2(1)*HN1(K)+LMBD2(2)*HN1(I)-A(2)                    MTR40105
        H2(I) = A(1)-LMBD1(1)*HN1(I)+LMBD1(2)*HN1(K)                    MTR40107
  110   H2(K) = A(2)-LMBD1(1)*HN1(K)-LMBD1(2)*HN1(I)                    MTR40108
      RETURN                                                            MTR40110
  200 DO  210  I=1,N                                                    MTR40112
        A(1)= LMBDN(1)*HN(I)                                            MTR40114
        H1(I) = LMBD2(1)*HN1(I)-A(1)                                    MTR40116
  210   H2(I) = A(1)-LMBD1*HN1(I)                                       MTR40117
      RETURN                                                            MTR40119
      END                                                               MTR40121
$      FORTRAN LSIOU,DECK
C AITKNS
      SUBROUTINE AITKNS (HN,HN1,HN2,HNEW,R,N,MAXR,NC,NP,IR)             MTR40123
      DIMENSION  HN(1), HN1(1), HN2(1), HNEW(1), A(2), B(4), C(2), D(2) MTR40125
      IR=0                                                              MTR40127
      L=NP*N                                                            MTR40129
```

```
      C(1) = R**2                                                        MTR40130
      DO  110  I=1,L,NP                                                  MTR40131
        D(1) = 0                                                         MTR40133
        D(2) = 0                                                         MTR40134
        DO  100  J=1,NC                                                  MTR40136
          K=(J-1)*MAXR+I                                                 MTR40138
          D(1) = D(1) + (HN1(K)-HN2(K))**2                               MTR40140
  100     D(2) = D(2) + (HN(K)-HN1(K))**2                                MTR40141
          IF  ( D(1) )                          105,110,105              MTR40143
  105   CONTINUE                                                         MTR40144
        IF  (D(2)/D(1) - C(1))                  110,110,800              MTR40145
  110   CONTINUE                                                         MTR40145
      GO TO  (300,200),NP                                                MTR40146
  200 CALL AITKND  (HN,HN1,HN2,HNEW,N,MAXR,NC,A,B,C,D)                   MTR40148
      GO TO  700                                                         MTR40149
  300 DO  600  I=1,N                                                     MTR40150
        GO  TO  (400,500),NC                                             MTR40152
  400   C(1) =  HN(I)-2.*HN1(I)+HN2(I)                                   MTR40154
        IF  ( C(1) )                            410,600,410              MTR40155
  410   HNEW(I) =  HN2(I) - ( (HN1(I)-HN2(I))**2 /C(1) )                 MTR40156
        GO TO 600                                                        MTR40158
  500   A(1) = 0.                                                        MTR40159
        C(1) = 2.                                                        MTR40161
        D(1) = 0.                                                        MTR40162
        DO  510  J=1,2                                                   MTR40163
          K = I+(J-1)*MAXR                                               MTR40164
          B(J) = HN(K)-2.*HN1(K)+HN2(K)                                  MTR40166
          C = C*(HN1(K)-HN2(K))                                          MTR40167
          A = -(HN1(K) - HN2(K))**2 -A                                   MTR40168
  510     D = B(J)**2 +D                                                 MTR40169
        IF  ( D )                               520,600,520              MTR40170
  520   HNEW(I) = HN2(I) - (B(1)*A(1)+B(2)*C(1))/ D(1)                   MTR40172
        HNEW(K) = HN2(K) - (B(1)*C(1)-B(2)*A(1))/ D(1)                   MTR40173
  600   CONTINUE                                                         MTR40174
  700   IR = 1                                                           MTR40176
  800 RETURN                                                             MTR40178
      END                                                                MTR40180
$       FORTRAN LSTOU.DECK                                               MTR40182
C AITKND
      SUBROUTINE  AITKND  (HN,HN1,HN2,HNEW,N,MAXR,NC,A,B,C,D)            MTR40184
      DIMENSION  HN(1), HN1(1), HN2(1),HNEW(1), A(1), B(2)  C(1), D(1)   MTR40186
      DOUBLE PRECISION  HN, HN1, HN2, HNEW, A, B, C, D                   MTR40187
  300 DO  600  I=1,N                                                     MTR40189
        GO  TO  (400,500),NC                                             MTR40191
  400   C(1) =  HN(I)-2.*HN1(I)+HN2(I)                                   MTR40192
      IF ( ABS(C(1)) -.000000100 )            600,600,410                MTR40193
  410   HNEW(I)=HN2(I) - ( (HN1(I)-HN2(I))*(HN1(I)-HN2(I)) /C(1) )       MTR40195
        GO TO 600                                                        MTR40196
  500   A(1) = 0.                                                        MTR40198
        C(1) = 2.                                                        MTR40199
        D(1) = 0.                                                        MTR40200
        DO  510  J=1,2                                                   MTR40201
          K = I+(J-1)*MAXR                                               MTR40203
          B(J) = HN(K)-2.*HN1(K)+HN2(K)                                  MTR40204
          C = C*(HN1(K)-HN2(K))                                          MTR40205
          A = -( (HN1(K)-HN2(K))*(HN1(K)-HN2(K)) ) - A                   MTR40206
  510     D = B(J)*B(J) + D                                              MTR40207
        IF  ( D )                               520,600,520              MTR40209
  520   HNEW(I) = HN2(I) - (B(1)*A(1)+B(2)*C(1))/ D(1)                   MTR40210
        HNEW(K) = HN2(K) - (B(1)*C(1)-B(2)*A(1))/ D(1)                   MTR40211
  600   CONTINUE                                                         MTR40213
```

```
700   IR = 1                                                          MTR40215
800 RETURN                                                            MTR40217
    END                                                               MTR40219
$      FORTRAN LSTOU.DECK
C LEGISS
      SUBROUTINE LEGIS (LAMBDA,POS,E1,E2,NC,IR)                       MTR40222
      DIMENSION  LAMBDA(1), POS(1)                                    MTR40224
      DOUBLE PRECISION  LAMBDA, POS, R1, R2A, R2B                     MTR40225
      IR=0                                                            MTR40227
      GO TO  (100,200),NC                                             MTR40229
  100 R1 = ( DABS( LAMBDA(1)-LAMBDA(2) ) )/E1                         MTR40231
      R2A =  DABS( DABS(POS(2)/POS(1))-1.) / E2                       MTR40232
      R2A =  DABS( DABS(POS(4)/POS(3))-1.) / E2                       MTR40233
      CALL MPRINT  (R1,2,1,2,6)                                       MTR40234
      CALL MPRINT  (R2A,2,1,2,6)                                      MTR40235
      CALL MPRINT  (R2B,2,1,2,6)                                      MTR40236
  110 IF  ( R1-R2A )                        120,140,140               MTR40238
  120 IF  ( R1-R2B )                        130,140,140               MTR40240
  130 IR=NC                                                           MTR40242
  140 RETURN                                                          MTR40244
  200 R1 = DSQRT( (LAMBDA(1)-LAMBDA(3))*(LAMBDA(1)-LAMBDA(3)) + (LAMBDA MTR40245
     1        (2)-LAMBDA(4))*(LAMBDA(2)-LAMBDA(4)) ) / E1              MTR40247
      R2A = DABS( DSQRT( ((POS(3)*POS(1)+POS(2)*POS(4))*(POS(3)*POS(1) +MTR40248
     1         POS(2)*POS(4)) + (POS(3)*POS(2)-POS(4)*POS(1))*(POS(3)*  MTR40249
     2         POS(2)-POS(4)*POS(1)) ) / (POS(1)*POS(1)+POS(2)*POS(2)) ) MTR40250
     3         -1. ) / E2                                              MTR40251
      R2B = DABS( DSQRT( ( (POS(7)*POS(5)+POS(6)*POS(8)) / (POS(5)*POS  MTR40253
     1        (5) + POS(6)*POS(6)) )*( (POS(7)*POS(5)+POS(6)*POS(8)) /  MTR40254
     2        (POS(5)*POS(5) + POS(6)*POS(6)) ) + ( (POS(7)*POS(6) - POS(MTR40255
     3        8)*POS(5)) / (POS(5)*POS(5) + POS(6)*POS(6)) ) * ( (POS(7) MTR40256
     4        *POS(6) - POS(8)*POS(5)) / (POS(5)*POS(5) + POS(6)*POS(6)) MTR40257
     5        ) ) -1.) / E2                                            MTR40258
      GO TO 110                                                       MTR40260
      END                                                             MTR40262
$      FORTRAN LSTOU.DECK
C MADZ
      SUBROUTINE  MADS (A,B,C,NSIZE,NC)                               MTR40264
      DOUBLE PRECISION  A(1), B(1), C(1)                              MTR40265
      K=NC*NSIZE                                                      MTR40267
      DO  100  I=1,K                                                  MTR40269
  100   C(I)=A(I)-B(I)                                                MTR40271
      RETURN                                                          MTR40272
      END                                                             MTR40273
$      FORTRAN LSTOU.DECK
C MULTS
      SUBROUTINE MULT  (A,B,C,LIZ,NIZ,MIZ,MAXA,MAXB,MAXC,NC,NP )      MTR40275
      DIMENSION  A(1), B(1), C(1)                                     MTR40277
      KA=NC*MAXA                                                      MTR40279
      KB=NC*MAXB                                                      MTR40280
      KC=NC*MAXC                                                      MTR40281
      GO TO (200,100),NP                                              MTR40283
  100 CALL   DMULT (A,B,C,LIZ,NIZ,MIZ,KA,KB,KC,NC)                    MTR40285
      GO TO 700                                                       MTR40286
  200 DO   600  L=1,LIZ                                               MTR40288
        DO   500  M=1,MIZ                                             MTR40290
          K= (M-1)*KB +1                                              MTR40292
          I= (M-1)*KC +L                                              MTR40293
          C(I)=0.                                                     MTR40295
          GO TO  (300,400),NC                                         MTR40297
  300       DO  310  N=1,NIZ                                          MTR40299
              J=(N-1)*KA+L                                            MTR40301
```

```
             JB = (N-1)+K                                              MTR40302
  310        C(I)=C(I)+A(J)*B(JB)                                      MTR40303
         GO TO 500                                                     MTR40304
  400    IC = I+MAXC                                                   MTR40306
         C(IC) =0.                                                     MTR40307
         DO  410  N=1,NIZ                                              MTR40309
           J= (N-1)*KA+L                                               MTR40311
           JB= (N-1) +K                                                MTR40312
           JC= J+MAXA                                                  MTR40313
           JBC= JB+MAXB                                                MTR40314
           C(I)=C(I)+A(J)*B(JB)-A(JC)*B(JBC)                           MTR40316
  410      C(IC)=C(IC)+A(J)*B(JBC)+A(JC)*B(JB)                         MTR40317
  500    CONTINUE                                                      MTR40319
  600  CONTINUE                                                        MTR40321
  700 RETURN                                                           MTR40323
      END                                                              MTR40325
$       FORTRAN LSTOU,DECK
C DMULTS
      SUBROUTINE  DMULT (A,B,C,LIZ,NIZ,MIZ,KA,KB,KC,NC)                MTR40327
      DOUBLE PRECISION  A(1), B(1), C(1)                               MTR40329
      MAXA=KA/2                                                        MTR40330
      MAXB=KB/2                                                        MTR40331
  200 DO  600  L=1,LIZ                                                 MTR40332
        DO  500  M=1,MIZ                                               MTR40334
         K= (M-1)*KB +1                                                MTR40336
         I= (M-1)*KC +L                                                MTR40337
         C(I)=0.                                                       MTR40339
         GO TO  (300,400),NC                                           MTR40341
  300      DO  310  N=1,NIZ                                            MTR40343
             J=(N-1)*KA+L                                              MTR40345
             JB = (N-1)+K                                              MTR40346
  310        C(I)=C(I)+A(J)*B(JB)                                      MTR40347
         GO TO 500                                                     MTR40348
  400    IC=I+KC/2                                                     MTR40350
         C(IC) =0.                                                     MTR40351
         DO  410  N=1,NIZ                                              MTR40353
           J= (N-1)*KA+L                                               MTR40355
           JB= (N-1) +K                                                MTR40356
           JC= J+MAXA                                                  MTR40357
           JBC = JB+MAXB                                               MTR40358
           C(I)=C(I)+A(J)*B(JB)-A(JC)*B(JBC)                           MTR40360
  410      C(IC)=C(IC)+A(J)*B(JBC)+A(JC)*B(JB)                         MTR40361
  500    CONTINUE                                                      MTR40363
  600  CONTINUE                                                        MTR40365
  700 RETURN                                                           MTR40367
      END                                                              MTR40369
$       FORTRAN LSTOU,DECK
C POLMS
      SUBROUTINE POLM (PN,PN1,QN,QN1,E2,LMBD1,LMBD2,NC,IR,IGO)         MTR40372
      DIMENSION  PN(1), PN1(1), QN(1), QN1(1), E2(1), LMBD1(1), LMBD2(1)MTR40374
      DOUBLE PRECISION  PN, PN1, QN, QN1,     LMBD1, LMBD2, A(2)       MTR40375
      IR=0                                                             MTR40376
      GO TO  (100,100,180),IGO                                         MTR40377
  100 GO TO  (110,200),NC                                              MTR40379
  110 IF ( DABS( (PN-PN1)/(DSQRT( DABS(QN) )) ) -E2 )     120,120,112  MTR40381
  112 IF ( DABS( PN-PN1 ) - E2**2 )                       120,120,170  MTR40382
  120 IF ( DABS( (QN/QN1)-1.) - E2 )                      130,130,170  MTR40384
  130 LMBD2 = (PN*PN -4.*QN)                                           MTR40386
      LMBD2 = DSQRT (DABS(LMBD2) )                                     MTR40387
      LMBD1 =(-PN + LMBD2) /2.                                         MTR40388
      LMBD2 =(-PN - LMBD2) /2.                                         MTR40389
```

```
      IF  ( ABS(LMBD1) - ABS(LMBD2) )                  140,150,160     MTR40391
  140 A= LMBD1                                                         MTR40392
      LMBD1=LMBD2                                                      MTR40393
      LMBD2=A                                                          MTR40394
      GO TO 160                                                        MTR40395
  150 IF  ( LMBD1 )                                    140,160,160     MTR40396
  160 IR=NC                                                            MTR40398
  170 RETURN                                                           MTR40400
  180 GO TO (130,220),NC                                               MTR40401
  200 A=DSQRT( QN(1)*QN(1) + QN(2)*QN(2) )                             MTR40403
      IF  ( DSQRT( ( (PN-PN1)*(PN-PN1) + (PN(2)-PN1(2))*(PN(2)-PN1(2)) )MTR40405
     1        / A ) -E2)                               210,210,170     MTR40406
  210 A = QN1*QN1 + QN1(2)*QN1(2)                                      MTR40407
      IF  ( ( ( (QN*QN1-QN(2)*QN1(2))/A )*( (QN*QN1-QN(2)*QN1(2))/A ) + MTR40409
     1        ( (QN*QN1(2)+QN1*QN(2))/A )*( (QN*QN1(2)+QN1*QN(2))/A )   MTR40410
     2     -1. ) - E2)                                 220,220,170     MTR40411
  220 A(1) = PN*PN - PN(2)*PN(2) -4.*QN                                MTR40412
      A(2) = 2.*(PN*PN(2)-2.*QN(2))                                    MTR40413
      IF  ( A(1) )                  230,250,250                        MTR40415
  230 A(1) = DSQRT( (-A(1)+DSQRT( A(1)*A(1)+A(2)*A(2) ) )  /2. )       MTR40417
      A(2) = A(2) / (2.*A(1))                                          MTR40418
  232 LMBD1(1) = (-PN+A(2))/2.                                         MTR40420
      LMBD1(2) = (-PN(2)+A(1))/2.                                      MTR40421
      LMBD2(1) = (-PN-A(2))/2.                                         MTR40422
      LMBD2(2) = (-PN(2)-A(1))/2.                                      MTR40423
      IF  ( DSQRT( LMBD1*LMBD1+LMBD1(2)*LMBD1(2) ) - DSQRT( LMBD2*LMBD2 +MTR40425
     1        LMBD2(2)*LMBD2(2)) )                     240,160,160     MTR40426
  240 A=LMBD1(2)                                                       MTR40427
      LMBD1(2)=LMBD2(2)                                                MTR40428
      LMBD2(2)= A                                                      MTR40429
      GO TO 140                                                        MTR40430
  250 A(1) = DSQRT( (A(1)+DSQRT( A(1)*A(1)+A(2)*A(2) )) /2. )          MTR40432
      A(2) = A(2) / (2.*A(1))                                          MTR40433
      LMBD1=A(1)                                                       MTR40434
      A(1)=A(2)                                                        MTR40435
      A(2)=LMBD1                                                       MTR40436
      GO TO 232                                                        MTR40437
      END                                                              MTR40439
       FORTRAN LSTOU,DECK
C POS
      SUBROUTINE  PQ  (FL, HN, HN1, HN2, HN3, P, Q, NC, MAXR )         MTR40441
      DIMENSION  FL(1), HN(1), HN1(1), HN2(1), HN3(1), P(1), Q(1),     MTR40442
     1          A(3), B(3)                                             MTR40443
      DOUBLE PRECISION  FL, HN, HN1, HN2, HN3, P, Q, A, B              MTR40445
      MAXR = MAXR/2                                                    MTR40447
      GO TO  (200,100), NC                                             MTR40448
  100 A(1) = FL(1)*HN  - FL(2)*HN(MAXR+1)                              MTR40450
      A(2) = FL(3)*HN1 - FL(4)*HN1(MAXR+1)                             MTR40451
      A(3) = FL(5)*HN2 - FL( 6)*HN2(MAXR+1)                            MTR40452
      B(1) = FL(1)*HN(MAXR+1)+ FL(2)*HN                                MTR40454
      B(2) = FL(3)*HN1(MAXR+1)+FL(4)*HN1                               MTR40455
      B(3) = FL(5)*HN2(MAXR+1)+FL( 6)*HN2                              MTR40456
      P(1) = A(3)*HN1 - B(3)*HN1(MAXR+1)-A(1)*HN3 + B(1)*HN3(MAXR+1)   MTR40458
      P(2) = A(3)*HN1(MAXR+1)+B(3)*HN1 - A(1)*HN3(MAXR+1)-B(1)*HN3     MTR40459
      Q(1) = A(1)*HN2 - B(1)*HN2(MAXR+1)-A(2)*HN1 + B(2)*HN1(MAXR+1)   MTR40461
      Q(2) = A(1)*HN2(MAXR+1)+B(1)*HN2 - A(2)*HN1(MAXR+1)-B(2)*HN1     MTR40462
      A(1) = A(2)*HN3 - B(2)*HN3(MAXR+1)-A(3)*HN2 + B(3)*HN2(MAXR+1)   MTR40464
      B(1) = A(2)*HN3(MAXR+1)+B(2)*HN3 - A(3)*HN2(MAXR+1)-B(3)*HN2     MTR40465
      A(2) = A(1)*P(1) + B(1)*P(2)                                     MTR40467
      B(2) = A(1)*P(2) - B(1)*P(1)                                     MTR40468
      A(3) = A(1)*Q(1) + B(1)*Q(2)                                     MTR40469
```

```
      B(3) = A(1)*Q(2) - B(1)*Q(1)                                      MTR40470
      A(1) = A(1)*A(1) + B(1,*B(1)                                      MTR4047
      A(2) = A(2)/A(1)                                                  MTR4047
      B(2) = B(2)/A(1)                                                  MTR40474
      A(3) = A(3)/A(1)                                                  MTR4047
      B(3) = B(3)/A(1)                                                  MTR4047
      P(1) = FL(3)*A(2) - FL(4)*B(2)                                    MTR40478
      P(2) = FL(3)*B(2) + FL(4)*A(2)                                    MTR40479
      A(2) = FL(3)*FL(5) - FL(4)*FL( 6)                                 MTR4048
      B(2) = FL(4)*FL(5) + FL(3)*FL( 6)                                 MTR40482
      Q(1) = A(2)*A(3)-B(2)*B(3)                                        MTR40484
      Q(2) = A(2)*B(3)+A(3)*B(2)                                        MTR4048
      MAXR = 2*MAXR                                                     MTR4048
      RETURN                                                            MTR40488
  200 A(1) = (FL(2)*HN1*HN3 - FL(3)*HN2*HN2)                            MTR4049
      P(1) = (FL(3)*HN2(1)*HN1(1) - FL(1)*HN(1)*HN3(1)) / A(1)          MTR4049
      Q(1) = ( FL(1)*HN(1)*HN2(1) - FL(2)*HN1(1)*HN1(1) )  / A(1)       MTR40492
      P(1) =  P(1)*FL(2)                                                MTR40494
      Q(1) =  Q(1)*FL(2)*FL(3)                                          MTR4049
      MAXR = 2*MAXR                                                     MTR4049
      RETURN                                                            MTR40499
      END                                                               MTR4050
$      FORTRAN LSTOU.DECK
C CLOSES
C  SUBROUTINE  CLOSE.        COMPUTES 2 CLOSE ROOTS.                    MTR4050
C                                                                       MTR4050
C         U = MATRIX. DIMENSIONED  (MAXR,2*NC*MAXR)                     MTR40505
C         H = STARTING GUESS, DIMENSIONED((MAXR,2*NC*4)+2*NC*N)         MTR40506
C         NSIZE = SIZE OF MATRIX                                        MTR4050
C         MAXR = DIMENSIONED NUMBER OF ROWS OF U AND H                  MTR4050
C         MAXTRY = MAXIMUM NUMBER OF DOUBLE PRECISION ITERATIONS.       MTR40509
C         EPS1 = SINGLE ROOT CONVERGENCE CRITERIA                       MTR4051
C         EPS2 = DOUBLE ROOT CONVERGENCE CRITERIA                       MTR4051
C         R= AITKENS CONVERGENCE CRITERIA                               MTR40512
C         IRR = ERROR RETURN INDICATER.    =1, OVERFLOW                 MTR40513
C                                          =2, DIVIDE CHECK             MTR4051
C                                          =3, BOTH OVERFLOW AND DIVIDE MTR4051
C                                                               CHECK.  MTR40516
C         ITERS= NUMBER OF ITERATIONS PERFORMED, - FOR DOUBLE ROOT      MTR4051
C                                                + FOR SINGLE ROOT      MTR4051
C         NC = 1, REAL      2, COMPLEX                                  MTR40519
      SUBROUTINE CLOSES (U,H,NSIZE,MAXR,R,EPS1,EPS2,NC,IRR,MAXTRY,ITERS,MTR40521
     1                  NAITKN,INDEX1,INDEX2,VALUE,MSIZE)               MTR4052
      DIMENSION   U(1),  H(1),  VALUE(1)                                MTR40524
      CALL  OVERFL ( IOVFLW )                                           MTR40526
      CALL  DVCHK ( IDVDCT )                                            MTR4052
      IRR=0                                                             MTR4052
      NX=2*NSIZE                                                        MTR40529
      N2C=2*NC                                                          MTR4053
      CALL  CHANGE   (U,MSIZE,NC*MSIZE,MA  ,1)                          MTR4053
      CALL  CHANGE   (H,NSIZE,NC,NSIZE,1)                               MTR40533
      I6=N2C*NSIZE                                                      MTR40535
      I1=1                                                              MTR4053
      I2=I1+I6                                                          MTR4053
      I3=I2+I6                                                          MTR40538
      I4=I3+I6                                                          MTR4053
      I5=I4+I6                                                          MTR4054
      K1=I1                                                             MTR40541
      K4=I2                                                             MTR40542
      K3=I3                                                             MTR4054
      K2=I4                                                             MTR40544
```

```
      ITERS = 0                                                         MTR40545
  100 K=0                                                               MTR40547
      IGO = 1                                                           MTR40548
  120 ITERS=ITERS+1                                                     MTR40549
      IF  ( ITERS-MAXTRY )                      140,140,130             MTR40550
  130 IF  ( K-0 )                               132,140,132             MTR40551
  132 ITERS = ITERS-1                                                   MTR40552
      GO TO 800                                                         MTR40553
  140 K=K+1                                                             MTR40555
      I=K1                                                              MTR40556
      K1=K4                                                             MTR40557
      K4=K3                                                             MTR40558
      K3=K2                                                             MTR40559
      K2=I                                                              MTR40560
      CALL MULT  (0      ,H(K2),H(K1),NSIZE,NSIZE,1,MAXR,NSIZE,NSIZE,   MTR40562
     1              NC,2)                                               MTR40563
      INDEX=0                                                           MTR40564
      IK=I5+(4-K)*N2C                                                   MTR40565
      CALL NORM  (H(K1),H(K1),NSIZE,H(IK),INDEX,NSIZE,NC,2)             MTR40566
      CALL  OVERFL ( IOVFLW )                                           MTR40568
      GOTO  (150,160) ,IOVFLW                                           MTR40569
  150 IRR=IRR+1                                                         MTR40570
  160 CALL  DVCHK ( IDVDCT )                                            MTR40571
      GOTO  (170,180) ,IDVDCT                                           MTR40572
  170 IRR=IRR+2                                                         MTR40573
  180 IF  ( IRR )                 200,200,640                           MTR40574
C   TEST FOR CONVERGENCE TO A SINGLE ROOT                               MTR40576
  200 DO  220  I=1,16,2                                                 MTR40578
        J2  = K2+I-1                                                    MTR40579
        J3  = K1+I-1                                                    MTR40580
          IF  ( ABS(H(J2)-H(J3))-EPS1)    220,220,300                   MTR40581
  220     CONTINUE                                                      MTR40582
      GOTO  750                                                         MTR40583
  300 GOTO  (120,120,120,320),K                                         MTR40585
  320 J1=I5+(IGO+3)*N2C                                                 MTR40587
      J2=I5+(IGO+5)*N2C                                                 MTR40588
      J=2*INDEX-2                                                       MTR40590
      J3=K1+J                                                           MTR40591
      J5=K2+J                                                           MTR40592
      J7=K3+J                                                           MTR40593
      J9=K4+J                                                           MTR40594
C     COMPUTE   P N  AND  Q N.                                          MTR40596
      CALL  PQ (H(I5), H(J3), H(J5), H(J7), H(J9), H(J1), H(J2), NC, NX)MTR40597
      GOTO (350,340),IGO                                                MTR40600
  340 J1= I5+4*N2C                                                      MTR40601
      J2= J1+N2C                                                        MTR40602
      J3= J2+N2C                                                        MTR40603
      J4= J3+N2C                                                        MTR40604
      J5= I5+N2C                                                        MTR40605
      J6= J5+N2C                                                        MTR40606
C TEST FOR DOUBLE ROOT CONVERGENCE AND IF SO, COMPUTE LAMBDA 1 AND 2.   MTR40608
      CALL POLM (H(J2),H(J1),H(J4),H(J3),EPS2,H(J5),H(J6),NC,IR,IGO)    MTR40610
      GOTO  (344,344,400),IGO                                           MTR40611
  344 IF  ( IR )                  346,346,400                           MTR40612
  346 IF  ( ITERS-MAXTRY )                 347,800,800                  MTR40613
  347 DO  348  I=1,N2C                                                  MTR40614
        H(J1)=H(J2)                                                     MTR40615
        H(J3)=H(J4)                                                     MTR40616
        J1=J1+1                                                         MTR40617
        J2=J2+1                                                         MTR40618
        J3=J3+1                                                         MTR40619
```

```
348    J4=J4+1                                                               MTR40620
350  K=3                                                                     MTR40622
     IGO=2                                                                   MTR40623
     DO  354  J=1,3                                                          MTR40624
     L=I5+(3-J)*N2C                                                          MTR40625
     L1=I5+(4-J)*N2C                                                         MTR40626
       DO  354  I=1,NC                                                       MTR40627
         L=L+(I-1)*NC                                                        MTR40628
         L1=L1+(I-1)*NC                                                      MTR40629
         H(L1)=H(L)                                                          MTR40630
354      H(L1+1)=H(L+1)                                                      MTR40631
     CALL  AITKNS (H(K1), H(K2), H(K3), H(K4), R,   ..E, NSIZE, NC, 2,      MTR40633
    1              IR    )                                                   MTR40634
     IF   ( IR )                360,120,360                                  MTR40635
360  I=K1                                                                    MTR40637
     K1=K4                                                                   MTR40638
     K4=K3                                                                   MTR40639
     K3=K2                                                                   MTR40640
     K2=I                                                                    MTR40641
     NAITKN = NAITKN+1                                                       MTR40642
     GOTO   100                                                              MTR40643
400  CALL  POH (H(I5), H(J5), H(J6), H(K1), H(K2), H(K3), H(K4), NSIZE,     MTR40645
    1                 NC)                                                    MTR40646
     GOTO   (404,402),NC                                                     MTR40648
402  VALUE(2)=H(J5+2)                                                        MTR40649
     VALUE(4)=H(J6+2)                                                        MTR40650
404  INDEX=0                                                                 MTR40651
     VALUE(1)=H(J5)                                                          MTR40652
     VALUE(NC+1)=H(J6)                                                       MTR40653
     CALL  NORM  (H(K3),H(K3),NSIZE,H(J1),INDEX,NSIZE,NC,2)                  MTR40654
     INDEX1 = INDEX                                                          MTR40656
     ITERS = -ITERS                                                          MTR40657
     I=K4+2*INDEX-2                                                          MTR40658
     H(J1)=H(I)                                                              MTR40659
     H(J1+1)=H(I+1)                                                          MTR40660
     GOTO (420,410),NC                                                       MTR40661
410  I = I+NX                                                                MTR40662
     H(J1+2)=H(I)                                                            MTR40663
     H(J1+3)=H(I+1)                                                          MTR40664
420  INDEX=-INDEX                                                            MTR40666
     CALL  NORM  (H(K4),H(K4),NSIZE,H(J1),INDEX,NSIZE,NC,2)                  MTR40667
     IF   ( K -'2 )                       440,500,480                        MTR40668
440  J3=13                                                                   MTR40669
442  J4=14                                                                   MTR40670
450  K=1                                                                     MTR40671
460  DO   462   J=1,16                                                       MTR40672
       J1=J3+J-1                                                             MTR40673
       J2=J4+J-1                                                             MTR40674
462    H(J1)=H(J2)                                                           MTR40675
     GOTO   (520,510,514,600),K                                              MTR40676
470  J3=11                                                                   MTR40677
     GOTO   442                                                              MTR40678
480  IF   ( K3-13  )                      490,490,470                        MTR40679
490  K=3                                                                     MTR40680
     J3=11                                                                   MTR40681
     J4=13                                                                   MTR40682
     GOTO   460                                                              MTR40683
500  K=2                                                                     MTR40684
     J3=13                                                                   MTR40685
     J4=11                                                                   MTR40686
     GOTO 460                                                                MTR40687
```

```
  510 J3=I1                                                            MTR40688
  512 J4=I2                                                            MTR40689
      GOTO  450                                                        MTR40690
  514 J3=I3                                                            MTR40691
      GOTO  512                                                        MTR40692
  520 CALL MADS   (H(I1),H(I3),H(I2),NSIZE,NC)                         MTR40693
  600 CALL CHANGE (H(I1),NSIZE,2*NC,NSIZE,2)                           MTR40695
  604 J=NC*NSIZE+1                                                     MTR40697
      INDEX=0                                                          MTR40698
      CALL NORM  (H(J),H(J),NSIZE,H(I5),INDEX,NSIZE,NC,1)              MTR40700
      J=J+INDEX1-1                                                     MTR40701
      J1=J+(NC-1)*NSIZE                                                MTR40702
      H(J)=0.                                                          MTR40703
      H(J1)=0.                                                         MTR40704
      INDEX2 = INDEX                                                   MTR40705
  606 CALL OVERFL  ( IOVFLW )                                          MTR40707
      GOTO  (610,620) ,IOVFLW                                          MTR40708
  610 IRR=IRR+1                                                        MTR40709
  620 CALL DVCHK ( IDVDCT )                                            MTR40711
      GOTO  (630,640) ,IDVDCT                                          MTR40712
  630 IRR=IRR+2                                                        MTR40713
  640 CALL CHANGE  (H,NSIZE,NC*NSIZE,MAXR,2)                           MTR40715
  700 RETURN                                                           MTR40717
  750 IK=I5+(4-K)*N2C                                                  MTR40719
      VALUE(1) = H(IK)                                                 MTR40720
      IK=IK+2*(NC-1)                                                   MTR40721
      VALUE(NC)=H(IK)                                                  MTR40722
      INDEX1=INDEX                                                     MTR40723
      K=4                                                              MTR40724
      J3=I1                                                            MTR40725
      J4=K1                                                            MTR40726
      IF  ( K1-I1 )           460,600,460                              MTR40727
  800 IGO=3                                                            MTR40729
      INDEX1=INDEX                                                     MTR40730
      J=I5                                                             MTR40731
      J1 = I5+4*N2C                                                    MTR40732
  806 CALL  LEGIS  ( H(J),H(J1),EPS1,EPS2,NC,IR )                      MTR40733
      IF  ( IR )              340,340,750                              MTR40734
      END                                                              MTR40736
$       FORTRAN LSTC[illegible].DECK
C MITERS
C  A IS STORED IN CORE AT A.      (MAXR X NC*NP*NSIZE )                MTR40738
C  NTAPUT IS A UTILITY TAPE, FOR CHECK VECTORS IF DESIRED.             MTR40739
C  EPSP = EPSILON ONE = SINGLE PRECISION CONVERGENCE TEST NUMBER       MTR40740
C  EPDP = EPSILON TWO = DOUBLE PRECISION       ..        ..   ..       MTR40741
C  NC = 1, IF REAL                      NP = 1, IF SINGLE PRECISION    MTR40742
C       2, IF COMPLEX                      = 2, IF DOUBLE      ..      MTR40743
C  NGUESS = 0, IF FIRST GUESS IS TO BE A COLUMN OF ONES.               MTR40744
C  MODOUT = NO. OF MODES TO BE COMPUTED.                               MTR40745
C  NAKSR = NO. TIMES AITKNS ACCELERATION WAS USED IN SINGLE PRECISION. MTR40746
C  NAKDR = ..      ..      ..      ..      ..   ..  .. DOUBLE    ..  . MTR40747
C  MAXSR = MAXIMUM ITERATIONS ALLOWED IN SINGLE PRECISION.             MTR40748
C  MAXDR =     ..      ..         ..   .. DOUBLE     ..    .           MTR40749
C  IRR = ERROR RETURN   = 1, FOR OVERFLOW                              MTR40750
C                         2, FOR DIVIDE CHECK                          MTR40751
C                         3, FOR BOTH OVERFLOW AND DIVIDE CHECK        MTR40752
C  NSIZE = NO. OF ROWS AND COLUMNS OF A                                MTR40753
C RSP =  R, AITKENS ACCELERATION CONVERGENCE CONTROL FOR SINGLE PRECIS. MTR40754
C  RDP =  R  AITKENS ACCELERATION CONVERGENCE CONTROL FOR DOUBLE PRECIS.MTR40755
C  MAXR  =   DIMENSIONED NUMBER OF ROWS OF A AND GUESS                 MTR40756
      SUBROUTINE MITERS(A,GUESS,NGUESS,NSIZE,MODOUT,MAXR,NC,EPSP,EPDP,  MTR40758
```

```
     1                  NAKSR,NAKDR,MAXSR,MAXDR,RSP,RDP,IRR,H,VECTOR,    MTR40759
     2                  VALUE,NITER,NTAPUT,NTAPOT)                       MTR40760
      DIMENSION    A(1), GUESS(1), H(1), NITER(1), NAKSR(1), NAKDR(1),   MTR40763
     1             ATITLE(9), VECTOR(1), VALUE(1)                        MTR40764
      DATA  (ATITLE(I),I=1,9)   / 10H OVERFLOW, , 10H          ,         MTR40766
     1             10H   DIVIDE , 10H CHECK   , 10H       IN ,           MTR40767
     2             10H CLOSE    , 10H SWEEP   , 10HSUBROUTINE ,          MTR40768
     3             10H          /                                        MTR40769
   10 FORMAT    (1H0  5X, 6H MODE   13X, 11H EIGENVALUE   10X,           MTR40771
     1          24HITERATIONS  S.P.    D.P.  6X, 20H AITKENS S.P.   D.P. MTR40772
     2            ///  )                                                 MTR40773
   12 FORMAT    (1H  1I11,  2F19.8,  1I22,  1I7,  1I19,  1I7 )           MTR40774
   14 FORMAT    (1H  1I11,  1F29.8,  1I31,  1I7,  1I19,  1I7 )           MTR40775
   16 FORMAT    (1H0 1I6, 43H MODES ARE CORRECT, COMPUTATION TERMINATED.)MTR40776
   18 FORMAT    (15H0MODIFIED MODE   1I6, 22H IS CORRECT, TRUE MODE      MTR40777
     1           20H CANNOT BE COMPUTED.     )                           MTR40778
   20 FORMAT    (1H0  /// 1H0  46X, 14H EIGENVECTORS      ///    )       MTR40779
   22 FORMAT    (1H1 10X, 36H CHECK EIGENVALUES AND EIGENVECTORS     //  MTR40780
     1          (6E16.8)    )                                            MTR40781
   24 FORMAT    (5H MODE   1I4, 29H HAS NOT CONVERGED IN MAXIMUM         MTR40782
     1          12H ITERATIONS    )                                      MTR40783
   26 FORMAT    (37H1***** ERROR IN ITERATION SUBROUTINE       PA10 )    MTR40784
      CALL OVERFL ( IOVFLW )                                             MTR40785
      CALL DVCHK ( IDVDCT )                                              MTR40787
      MSIZE = NSIZE                                                      MTR40788
      ISIZE = NC*MAXR                                                    MTR40789
      IRR=0                                                              MTR40790
C FIND A AND STORE ON TAPE IF NECESSARY.                                 MTR40792
      IF  ( NTAPUT )          102,102,100                                MTR40793
  100 REWIND NTAPUT                                                      MTR40794
      M=NSIZE*ISIZE                                                      MTR40795
        WRITE   ( NTAPUT )     (A(I),I=1,M)                              MTR40796
      REWIND NTAPUT                                                      MTR40797
  102 IF  ( EPSP )             106,104,106                               MTR40799
  104 EPSP = .5E-08                                                      MTR40800
  106 IF  ( EPDP )             110,108,110                               MTR40802
  108 EPDP = .5E-08                                                      MTR40803
  110 JSIZE=NSIZE*(NC-1)                                                 MTR40805
C DEFINE FIRST GUESS,  IF NOT GIVEN.                                     MTR40807
      IF  ( NGUESS )           111,114,111                               MTR40809
  111 IF  ( MAX -NSIZE )       116,116,112                               MTR40810
  112 DO  113  I=1,NSIZE                                                 MTR40811
        J= NSIZE+I                                                       MTR40812
        J1= MAXR+I                                                       MTR40813
  113   GUESS(J)=GUESS(J1)                                               MTR40814
      GOTO  116                                                          MTR40815
  114 DO  115  I=1,NSIZE                                                 MTR40817
        J=JSIZE+I                                                        MTR40818
        GUESS(J) = 0.                                                    MTR40819
  115   GUESS(I)=1.                                                      MTR40820
C DEFINE PROGRAM CONSTANTS.                                              MTR40822
  116 MODE = 0                                                           MTR40824
      I1=1                                                               MTR40825
          I2 = I1+ISIZE                                                  MTR40826
          I3 = I2+ISIZE                                                  MTR40827
C SET FOR MODE COUNT                                                     MTR40829
  130 MODE=MODE+1                                                        MTR40831
      KSIZE = NC*MSIZE                                                   MTR40832
      IF  ( MODE- MODOUT )                 140,140,500                   MTR40833
  140 NSRAK=0                                                            MTR40835
```

```
      NORAK=0                                                           MTR40836
      ITERSR=0                                                          MTR40837
      ITERDR=0                                                          MTR40838
      K1=I1                                                             MTR40840
      K2=I3                                                             MTR40841
      K3=I2                                                             MTR40842
      DO  150  K=1,NC                                                   MTR40844
        J1=(K-1)*NSIZE                                                  MTR40845
        J= (K-1)*MSIZE+K1-1                                             MTR40846
        DO  150  I=1,MSIZE                                              MTR40847
          J=J+1                                                         MTR40848
          J1=J1+1                                                       MTR40849
  150     H(J)=GUESS(J1)                                                MTR40850
  152 NAK=0                                                             MTR40852
  160 NAK=NAK+1                                                         MTR40853
  162 ITERSR=ITERSR+1                                                   MTR40855
      IF  ( ITERSR-MAXSR )                  170,170,250                 MTR40857
  170 I=K1                                                              MTR40859
      K1=K3                                                             MTR40860
      K3=K2                                                             MTR40861
      K2=I                                                              MTR40862
C   SET .... NOW MAKE ONE ITERATION.                                    MTR40864
C                                                                       MTR40865
      CALL MULT (A,H(K2),H(K1),MSIZE,MSIZE,1,MAXR,MSIZE,MSIZE,NC,1)     MTR40866
      INDEX=0                                                           MTR40868
      IK= I3+KSIZE+NC*(3-NAK)                                           MTR40869
      CALL NORM (H(K1),H(K1),MSIZE,H(IK),INDEX,MSIZE,NC,1)              MTR40870
      CALL OVERFL  ( IOVFLW )                                           MTR40872
      CALL DVCHK  ( IDVDCT )                                            MTR40873
      GOTO  (180,182) ,IOVFLW                                           MTR40875
  180 IRR=IRR+1                                                         MTR40876
  182 GOTO  (184,186) ,IDVDCT                                           MTR40877
  184 IRR=IRR+2                                                         MTR40878
  186 IF  ( IRR )              190,190,600                              MTR40880
  190 GOTO  (160,200,200),NAK                                           MTR40882
  200 DO  210  I=1,KSIZE                                                MTR40884
        J1=K1+I-1                                                       MTR40885
        J2=K2+I-1                                                       MTR40886
        IF  ( ABS(H(J1)-H(J2))-EPSP )             210,210,220           MTR40887
  210   CONTINUE                                                        MTR40888
      GOTO  400                                                         MTR40889
  220 GOTO  (160,160,230),NAK                                           MTR40891
  230 CALL AITKNS ( H(K1), H(K2), H(K3), H(K3), RSP, MSIZE, MSIZE, NC,  MTR40893
     1                    1, IR )                                       MTR40894
      IF  ( IR )               240,232,240                              MTR40895
  232 L=2*NC                                                            MTR40896
      J=IK+L                                                            MTR40897
      DO  234  I=1,L,NC                                                 MTR40898
        J1=J+NC-I                                                       MTR40899
        J2=J-I+1                                                        MTR40900
        J3=J1-NC                                                        MTR40901
        J4=J2-NC                                                        MTR40902
        H(J1)=H(J3)                                                     MTR40903
  234   H(J2)=H(J4)                                                     MTR40904
      GOTO  162                                                         MTR40905
  240 I=K1                                                              MTR40907
      K1=K3                                                             MTR40908
      K3=K2                                                             MTR40909
      K2=I                                                              MTR40910
      NSRAK=NSRAK+1                                                     MTR40911
      GOTO  152                                                         MTR40912
```

```
250 CALL   OVERFL   ( IOVFLW )                                              MTR40914
    CALL   DVCHK   ( IDVDCT )                                               MTR40915
    GOTO   (260,262) ,IOVFLW                                                MTR40916
260 IRR=IRR+1                                                               MTR40917
262 GOTO   (264,266) ,IDVDCT                                                MTR40918
264 IRR=IRR+2                                                               MTR40919
266 IF   ( IRR   )                 270,270,600                              MTR40920
270 IF (K1-I1 )                    272,280,272                              MTR40922
272 DO  274   I=1,KSIZE                                                     MTR40924
      J=K1+I-1                                                              MTR40925
      J1=I1+I-1                                                             MTR40926
274   H(J1) = H(J)                                                          MTR40927
280 J=(MODE-1)*NC+1                                                         MTR40928
    ITERSR=ITERSR-1                                                         MTR40929
    CALL   CLOSES(A,H(I1),MSIZE,MAXR,RDP,EPSP,EPDP,NC,IRR,MAXDR,ITERDR,MTR40930
   1                     NDRAK,INDEX1,INDEX2,VALUE(J), NSIZE )              MTR40931
    IF   ( IRR )                   282,282,610                              MTR40945
282 IF   ( ITERDR )                283,288,288                              MTR40946
283 IF   ( KSIZE-ISIZE )           284,288,288                              MTR40948
284 J1  =  I1+2*KSIZE                                                       MTR40950
    J2  =  I2+KSIZE                                                         MTR40951
    DO   286   I=1,KSIZE                                                    MTR40952
      J1= J1-1                                                              MTR40953
      J2= J2-1                                                              MTR40954
286   H(J2)=H(J1)                                                           MTR40955
288 INDEX = INDEX1                                                          MTR40957
    INDIX = INDEX                                                           MTR40958
290 M1  = NSIZE-1                                                           MTR40959
    J1=INDEX                                                                MTR40960
    IF   ( J1-M1  )                          292,292,298                    MTR40961
292 DO   296   K=1,ISIZE                                                    MTR40962
      L=(K-1)*MAXR                                                          MTR40963
      L1=L+INDEX                                                            MTR40964
      HOLD= A(L1)                                                           MTR40965
      DO   294   J=J1,M1                                                    MTR40966
        I=L+J                                                               MTR40967
294     A(I) = A(I+1)                                                       MTR40968
296   A(I+1)=HOLD                                                           MTR40969
298   L=NSIZE-MODE+1                                                        MTR40970
    J = (MODE-1) * NC * MAXR +1                                             MTR40972
    CALL   SWEEPX (VECTOR(J),A, H, A(L), VALUE, MODE, MSIZE, MAXR, NC,      MTR40977
   1                  INDIX, EPSP, NSIZE, NITER(MODOUT+1) , IRR)            MTR40978
    CALL   OVERFL   ( IOVFLW )                                              MTR40979
    CALL   DVCHK    ( IDVDCT )                                              MTR40980
    GOTO   ( 300,302) ,IOVFLW                                               MTR40982
300 IRR=IRR+1                                                               MTR40983
302 GOTO   ( 304,306) ,IDVDCT                                               MTR40984
304 IRR=IRR+2                                                               MTR40985
306 IF   ( IRR )                   310,310,620                              MTR40986
310 L=(NC-1)*NSIZE                                                          MTR40988
    DO   312   J=INDEX,NSIZE                                                MTR40989
      L=L+J                                                                 MTR40990
      GUESS(L)=GUESS(L+1)                                                   MTR40991
312   GUESS(J)=GUESS(J+1)                                                   MTR40992
    MSIZE = MSIZE-1                                                         MTR40994
    NITER(MODE) = ITERSR+ITERDR                                             MTR40995
    NAKSR(MODE)= NSRAK                                                      MTR40996
    NAKDR(MODE)= NDRAK                                                      MTR40997
    IF   ( ITERDR )                320,360,360                              MTR40998
320 MODE=MODE+1                                                             MTR40999
    ITERDR=-ITERDR                                                          MTR41000
```

```
      NITER(MODE-1)= 0.                                                 MTR41001
      INDEX=INDEX2                                                      MTR41002
      IF   ( INDEX-NSIZE )                     326,330,330              MTR41003
  326 IF   ( INDEX1-INDEX2 )                   330,330,340              MTR41004
  330 INDIX=-INDEX                                                      MTR41005
     -INDEX=INDEX-1                                                     MTR41006
      GOTO 342                                                          MTR41007
  340 INDIX=-INDEX                                                      MTR41008
  342 CONTINUE                                                          MTR41009
      I1=I1+ISIZE                                                       MTR41010
      I2=I2+ISIZE                                                       MTR41011
      I3=I3+ISIZE                                                       MTR41012
      NAKSR(MODE-1) = 0.                                                MTR41013
      NAKDR(MODE-1) = 0.                                                MTR41014
      GOTO  290                                                         MTR41015
  360 I1=I1+ISIZE                                                       MTR41017
      I2=I1+ISIZE                                                       MTR41018
      I3=I2+ISIZE                                                       MTR41019
      GOTO  130                                                         MTR41020
  400 IF   ( K1-I1 )              410,413,410                           MTR41022
  410 DO   412   I=1,KSIZE                                              MTR41023
        J=K1+I-1                                                        MTR41024
        J1=I1+I-1                                                       MTR41025
  412   H(J1)=H(J)                                                      MTR41026
  413 DO   414   J=1,NC                                                 MTR41027
        I= NC*(MODE-1)+I                                                MTR41028
        J1=IK+J-1                                                       MTR41029
  414   VALUE(I)= H(J1)                                                 MTR41030
      INDIX=INDEX                                                       MTR41031
      GOTO  290                                                         MTR41032
  500 MODE=MODE-1                                                       MTR41034
      IF   ( MODE )                 530,530,502                         MTR41035
  502 IF   ( NTAPUT )               510,510,504                         MTR41036
  504 READ   ( NTAPUT )        (A(I),I=1,M)                             MTR41037
      CALL   MULT (A,VECTOR, H,NSIZE,NSIZE,MODE,MAXR,MAXR,MAXR ,NC,1)   MTR41039
      DO   506   I=1,MODE                                               MTR41041
        J= (I-1)*ISIZE    + 1                                           MTR41043
        J1 = (I-1)*NC+1                                                 MTR41044
        INDEX=0                                                         MTR41045
  506   CALL  NORM (H(J),H(J),NSIZE,GUESS(J1),INDEX,MAXR ,NC,1)         MTR41046
  510 IF   ( NTAPOT )               530,530,512                         MTR41048
  512 WRITE   (NTAPOT,10 )                                              MTR41050
      DO   522   I=1,MODE                                               MTR41051
        J1= MAXSR                                                       MTR41052
        J2= NITER(I)-MAXSR                                              MTR41053
        IF   ( J2 )                            514,515,515              MTR41054
  514   J1=NITER(I)                                                     MTR41055
        J2=0                                                            MTR41056
        GO TO   517                                                     MTR41057
  515   IF   ( J2-MAXDR )                      517,516,516              MTR41058
  516   WRITE (NTAPOT,24)          I                                    MTR41059
  517   GO TO   (518,520) ,NC                                           MTR41060
  518   WRITE (NTAPOT,11)          I, VALUE(I),  J1,J2, NAKSR(I),NAKDR(I) MTR41062
        GOTO   522                                                      MTR41063
  520   J= 2*I                                                          MTR41065
        WRITE (NTAPOT,12)  I, VALUE(J-1), VALUE(J), J1,J2, NAKSR(I),    MTR41066
     1                        NAKDR(I)                                  MTR41067
  522   CONTINUE                                                        MTR41068
      J=NC*MODE                                                         MTR41069
      WRITE   (NTAPOT,20)                                               MTR41070
      CALL   MPRINT    (VECTOR,NSIZE,J,MAXR,NTAPOT)                     MTR41071
```

```
      IF   ( NTAPUT )                         530,530,524
  524 WRITE  (NTAPOT,22)    ((GUESS(I),I=1,J))                        MTR41073
      CALL   MPRINT ( H,NSIZE,J,MAXR ,NTAPOT)                         MTR41077
  530 RETURN                                                          MTR41076
  600 NITER(MODE) = ITERSP+ITEROR                                     MTR41078
      NAKSR(MODE) = NSRAK                                             MTR41079
      NAKOR(MODE) = NORAK                                             MTR41080
      IF   ( NTAPOT )           500,600,601                           MTR41081
  601 J=1                                                             MTR41082
      J1=4                                                            MTR41084
      GOTO   (602,604,606),IRR                                        MTR41085
  602 J1=2                                                            MTR41086
      GOTO    606                                                     MTR41087
  604 J=3                                                             MTR41087
  606 WRITE   (NTAPOT,26)     (ATITLE(I),I=J,J1)                      MTR41088
      GOTO   630                                                      MTR41089
  610 J2=6                                                            MTR41090
  612 NITER(MODE) = ITERSP+ITEROR                                     MTR41091
      NAKSR(MODE) = NSRAK                                             MTR41092
      NAKOR(MODE) = NORAK                                             MTR41093
      IF   ( NTAPOT )           530,530,613                           MTR41094
  613 J=1                                                             MTR41095
      J1=4                                                            MTR41096
      GOTO   (614,616,618),IRR                                        MTR41097
  614 J1=2                                                            MTR41098
      GOTO   618                                                      MTR41099
  616 J=3                                                             MTR41100
  618 WRITE   (NTAPOT,26)    (ATITLE(I),I=J,J1),   (ATITLE(5)),       MTR41101
     1                  (ATITLE(J2)),   (ATITLE(I),I=8,9)             MTR41102
      IF   ( J2-6 )                             630,630,622           MTR41103
  620 J2=7                                                            MTR41104
      GOTO   612                                                      MTR41106
  622 WRITE   (NTAPOT,18)     MODE                                    MTR41107
      IRR=0                                                           MTR41110
      J= (MODE-1)*NC*MAXR                                             MTR41111
      DO   624   I=1,ISIZE                                            MTR41112
        J1=J+I                                                        MTR41113
        J2=J+(NC-1)*MAXR+I                                            MTR41114
        VECTOR(J2)=0.                                                 MTR41115
  624   VECTOR(J1)=0.                                                 MTR41116
      GOTO   310                                                      MTR41117
  630 I=MODE-                                                         MTR41118
      WRITE   (NTAPOT,16)      I                                      MTR41120
      GOTO   500                                                      MTR41121
      END                                                             MTR41122
$       FORTRAN LSTOU,DECK                                            MTR41124
C SWEEPS
C
C           S W E E P X     S U B R O U T I N E
C                                                                     MTR41126
C COMPUTES TRUE MODE AND SWEEPS IT FROM THE MATRIX. (REAL   R COMPLEX)  MTR41127
C                                                                     MTR41128
C  HTRUE  =  TRUE MODAL COLUMNS, AS COMPUTED. U = DYNAMIC MATRIX.      MTR41129
C      H  :  SERIES OF MODIFIED MODAL COLUMNS.FL= COLUMN OF EIGENVALUES. MTR41130
C     US  :  SERIES OF MODIFIED MODAL ROWS OF U.                      MTR41131
C   MODE  =  MODE NOW BEING COMPUTED.            N = SIZE             MTR41132
C     MD  :  DIMENSIONED NUMBER OF ROWS OF U,US,H,HTRUE               MTR41133
C     NX  :  1 IF PROBLEM IS REAL.                                    MTR41134
C         :  2 IF PROBLEM IS COMPLEX.                                 MTR41135
      SUBROUTINE   SWEEPX   (HTRUE, U, H, US, FL, MODE, N, MD, NC, INDIX, MTR41136
     1                       EP, MSIZE, INDIS, IRR)                   MTR41137
      DIMENSION    H(1), US(1), U(1), HTRUE(1), FL(1), G(4)   INDIS(1) MTR41138
```

```
      INDEX=IABS(INDIX)                                                 MTR41156
      INDIS(MODE) = INDEX                                               MTR41157
      M=MODE-1                                                          MTR41158
      K1=M*NC*MD                                                        MTR41159
      NN=NC*MD                                                          MTR41160
      K2=MSIZE-M                                                        MTR41161
      IF  ( M )                 70,70,80                                MTR41162
   70 GOTO  (140,72) ,NC                                                MTR41163
   72 IF  ( MSIZE-MD )          74,140,140                              MTR41164
   74 L=K1+2*N+1                                                        MTR41165
      K=K1+MD+N+1                                                       MTR41166
      DO  76  I=1,N                                                     MTR41167
        K=K-1                                                           MTR41168
        L=L-1                                                           MTR41169
        H(K)=H(L)                                                       MTR41170
   76   H(L)=0.                                                         MTR41171
      GOTO  140                                                         MTR41172
   80 DO  90  I=1,M                                                     MTR41173
        J1=MODE+I                                                       MTR41174
   90   INDIS(J1)=INDIS(I)                                              MTR41175
  100 IF  ( INDIX )             102,104,104                             MTR41176
  102 K2=K2+1                                                           MTR41177
      M=M-1                                                             MTR41178
  104 J1=K1+K2*NC+1                                                     MTR41179
      J2=K1+MSIZE*NC                                                    MTR41180
      IF  ( J1-J2 )                              105,105,107            MTR41181
  105 DO  106  I=J1,J2                                                  MTR41182
  106   H(I)=0.                                                         MTR41183
  107 GO TO (114,108),NC                                                MTR41184
  108 J1=K1+K2+MD+1                                                     MTR41185
      J2=K1+K2*NC+1                                                     MTR41186
      DO  110  I=1,K2                                                   MTR41187
        J1=J1-1                                                         MTR41188
        J2=J2-1                                                         MTR41189
  110   H(J1)=H(J2)                                                     MTR41190
      IF  ( M )                 118,116,111                             MTR41191
  111 DO  112  I=1,M                                                    MTR41192
        H(J2)=0.                                                        MTR41193
  112   J2=J2+1                                                         MTR41194
  114 IF  ( M )                              118,118,120                MTR41195
  120 II=1                                                              MTR41196
      DO  130  I=1,M                                                    MTR41197
  121   DO  122  J=1,M                                                  MTR41198
          J1=MODE+J                                                     MTR41199
          IF  ( II - INDIS(J1))              122,123,122                MTR41200
  122     CONTINUE                                                      MTR41201
        II=II+1                                                         MTR41202
        GOTO  121                                                       MTR41203
  123   INDIS(J1)=0                                                     MTR41204
        IF  ( INDIS(MODE)-II )               125,124,124                MTR41205
  124   INDIS(MODE)=INDIS(MODE)+1                                       MTR41206
  125   I=MSIZE-II                                                      MTR41207
        IF  ( I )               129,129,126                             MTR41208
  126   J1= K1+MSIZE                                                    MTR41209
        J2= K1   +MSIZE +(NC-1)*MD                                      MTR41210
        DO  128  J=1,I                                                  MTR41211
          J1=J1-1                                                       MTR41212
          J2=J2-1                                                       MTR41213
          H(J2+1) = H(J2)                                               MTR41214
  128     H(J1+1) = H(J1)                                               MTR41215
  129   J1=K1+II                                                        MTR41216
```

```
      J2=J1+(NC-1)*MD
      H(J1)=0.
      H(J2)=0.
  130 CONTINUE
      IF  ( INDIX )            118,131,131
  118 M=M+1
  131 DO  132  I=1,M
      J1=MODE+I
  132 INDIS(J1)=INDIS(I)
      II=1
      DO  138  I=1,M
  133  DO 1134  J=1,M
        J1=MODE+J
        IF  ( II-INDIS(J1) )           1134,1135,1134
 1134   CONTINUE
      II=II+1
      GOTO  133
 1135 L1=(NC*MSIZE-1)*MD+MODE+1
      L3=(MSIZE-II)*NN
      INDIS(J1) = 0
      DO  138  J=1,NC
        J2=L1-(J-1)*MD-1
        IF  ( L3-NN )              138,134,134
  134   DO  136  I=1,L3,NN
          J1=L1-I
          J2=J1-NN
  136     US(J1)=US(J2)
        L1=L1-MD
  138   US(J2)=0.
  140 DO  142  I=1,NN
      J1=K1+I
  142 HTRUE(I)=H(J1)
      IF  ( M )              31,31,8
    8 DO  25  I=1,M
      K=MODE-I
      L1=NC*MD*(K-1)
      CALL  MULT (US( K   ), HTRUE, G, 1, MSIZE, 1, MD, MD, 1, NC, 1 )
      IF  ( G(1) )                   12,9,12
    9 GO TO (11,10),NC
   10 IF  ( G(2) )                   12,11,12
   11 IRR=IRR+2
   12 CONTINUE
      GOTO (14,19),NC
   14 IF (ABS(FL(K)/FL(MODE)-1.0) - EP )          15,15,16
   15 G=1.0
      GOTO  17
   16 G=(FL(K)-FL(MODE)) / G
   17 DO  18  J=1,MSIZE
        I=L1+J
   18   HTRUE(J)=H(I)-G(1)*HTRUE(J)
      GOTO  25
   19 K=2*K
      J=2*MODE
      IF ( ABS((FL(K-1)*FL(J-1)+FL(K)*FL(J))/(FL(J-1)**2+FL(J)**2)-1.0)
     1        -EP)        20,20,22
   20 IF ( ABS((FL(K)*FL(J-1)-FL(K-1)*FL(J)) / (FL(J-1)**2+FL(J)**2))
     1        -EP)        21,21,22
   21 G(1)=1.0
      G(2)=0.0
      GOTO  23
   22 G(3)=G(1)**2+G(2)**2
```

```
      G(4)=(FL(K)-FL(J))*G(1)-(FL(K-1)-FL(J-1))*G(2)                    MTR41283
      G(1)=((FL(K-1)-FL(J-1))*G(1)+(FL(K)-FL(J))*G(2)) / G(3)           MTR41284
      G(2)= G(4) / G(3)                                                 MTR41285
 23   DO  24  J1=1,MSIZE                                                MTR41286
        K2=J1+MD                                                        MTR41287
        L=L1+J1                                                         MTR41288
        L2=L+MD                                                         MTR41289
        G(3)=HTRUE(J1)                                                  MTR41290
        HTRUE(J1) = H(L) + G(2)*HTRUE(K2) -G(1)*HTRUE(J1)               MTR41291
        HTRUE(K2)= H(L2)- G(1)*HTRUE(K2)-G(2)*G(3)                      MTR41292
 24     CONTINUE                                                        MTR41293
 25   CONTINUE                                                          MTR41296
      I=0                                                               MTR41297
      CALL  NORM ( HTRUE,HTRUE, MSIZE, G, I, MD, NC, 1)                 MTR41298
 31   J1 = 1                                                            MTR41299
      J2 = 1                                                            MTR41300
      L4 = MODE                                                         MTR41301
      DO  43    I=1,MSIZE                                               MTR41302
        L1=J1                                                           MTR41303
        L2=J2                                                           MTR41304
        L3=K1+1                                                         MTR41305
        DO  33   MM=1,MODE                                              MTR41306
          IF ( I -INDIS(MM) )                 33,39,33                  MTR41307
 33       CONTINUE                                                      MTR41308
        DO  37  J=1,MSIZE                                               MTR41309
          DO  35    MM=1,MODE                                           MTR41310
            IF   ( J-INDIS(MM) )              35,37,35                  MTR41311
 35         CONTINUE                                                    MTR41312
          U(L1) = U(L2) - H(L3)*US(L4)                                  MTR41313
          GOTO    (38,36),NC                                            MTR41314
 36       J3= L1+MD                                                     MTR41315
          J4= L2+MD                                                     MTR41316
          J5= L4+MD                                                     MTR41317
          J6= L3+MD                                                     MTR41318
          U(J3)=U(J4)-H(L3)*US(J5) - H(J6)*US(L4)                       MTR41319
          U(L1)= U(L1) + H(J6)*US(J5)                                   MTR41320
 38       CONTINUE                                                      MTR41321
          L1=L1+1                                                       MTR41322
          L2=L2+1                                                       MTR41323
 37       L3=L3+1                                                       MTR41324
        GOTO  41                                                        MTR41325
 39     IF   ( I-INDIS(MODE) )               43,42,43                   MTR41326
 41     J1=J1+NN                                                        MTR41327
 42     J2=J2+NN                                                        MTR41328
 43     L4=L4+NN                                                        MTR41329
      L4=(MSIZE-MODE)*NN+1                                              MTR41331
      DO  52  J=1,NC                                                    MTR41332
        L4 = L4 + (J-1)*MD                                              MTR41333
      L1=L4+(MSIZE-MODE)-1                                              MTR41334
      DO  52  I=L4,L1                                                   MTR41335
 52     U(I)=0.                                                         MTR41336
      RETURN                                                            MTR41342
      END                                                               MTR41343
$      FORTRAN LSTOU,DECK
C CHANGES
      SUBROUTINE   CHANGE  (A,M,N,MAXR,ICHUZ)                           MTR41345
      DIMENSION  A(1)                                                   MTR41346
      MB=2*MAXR                                                         MTR41347
      GO TO  (10,20) ,ICHUZ                                             MTR41348
   10 DO  12  I=1,N                                                     MTR41349
        K=(N-I)*MAXR+M+1                                                MTR41350
```

```
      KK=(N-1)*MB+2*M+1
      DO  12  J=1,M
        IGET=K-J
        IPUT=KK-2*J
        A(IPUT)=A(IGET)
 12     A(IPUT+1)=0.
    GO TO  100
 20 K=0
    KK=0
    DO  24  I=1,N
      DO  22  J=1,M
        IGET = KK+2*J-1
        IPUT=K+J
 22     A(IPUT) = A(IGET)
      K=K+MAXR
 24   KK=KK+MB
100 RETURN
    END
```

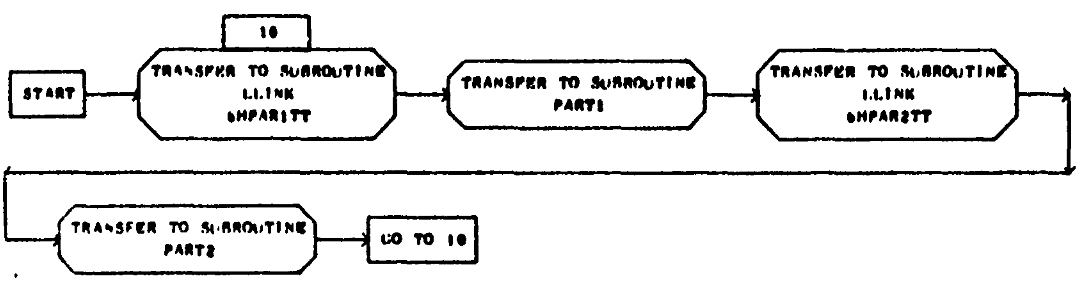
SECTION 9

FLOW DIAGRAMS

MAIN FLUTTER OVERLAY

JAN 11,1967

COMMON IT(218) PAGE 1

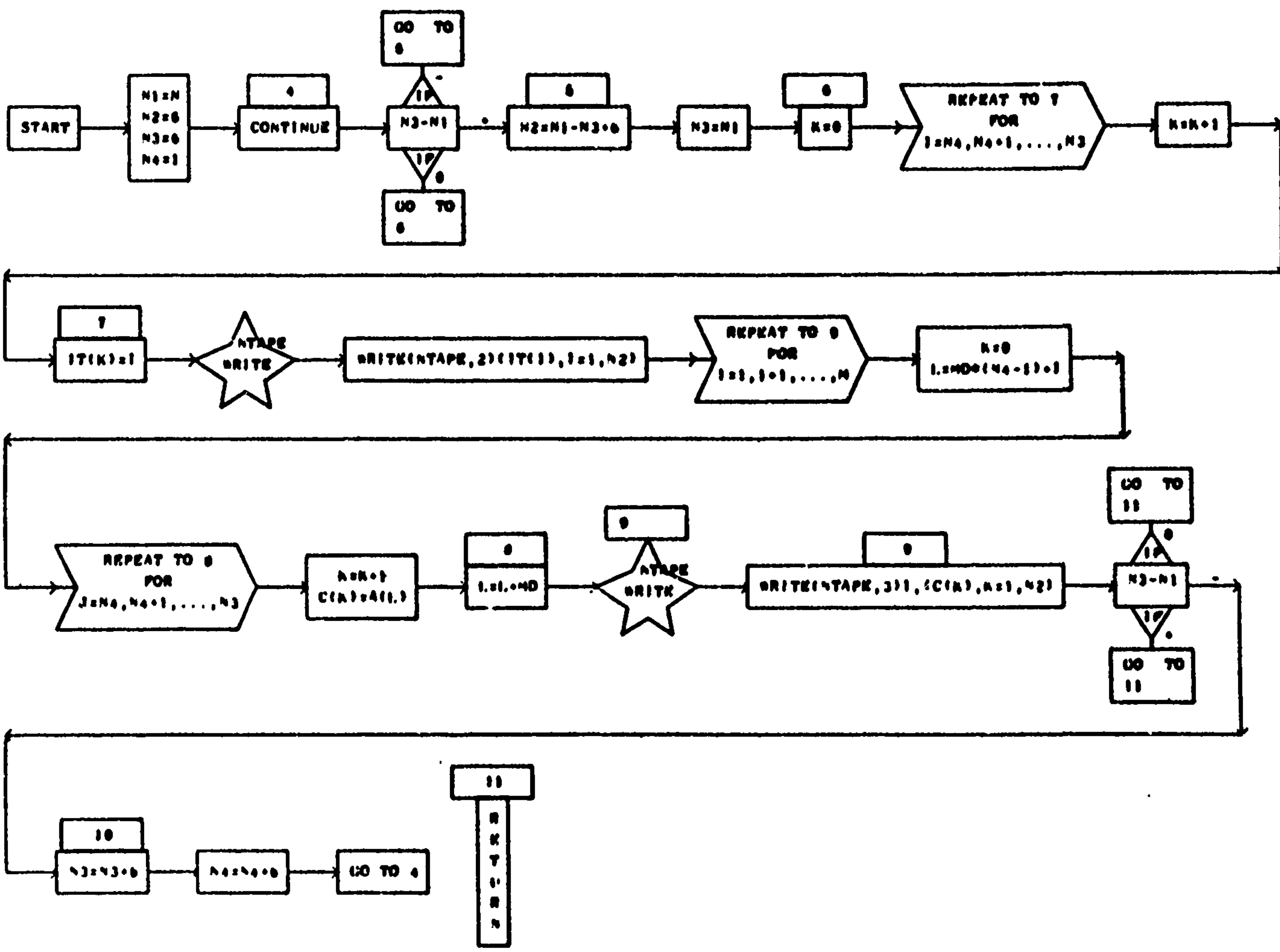


MPRINT

DIMENSIONED VARIABLES

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	IT	6	C	6				

SUBROUTINE MPRINT (A,M,N,ND,NTAPE) PAGE 1

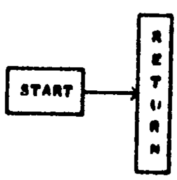


MPUNCH

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1								

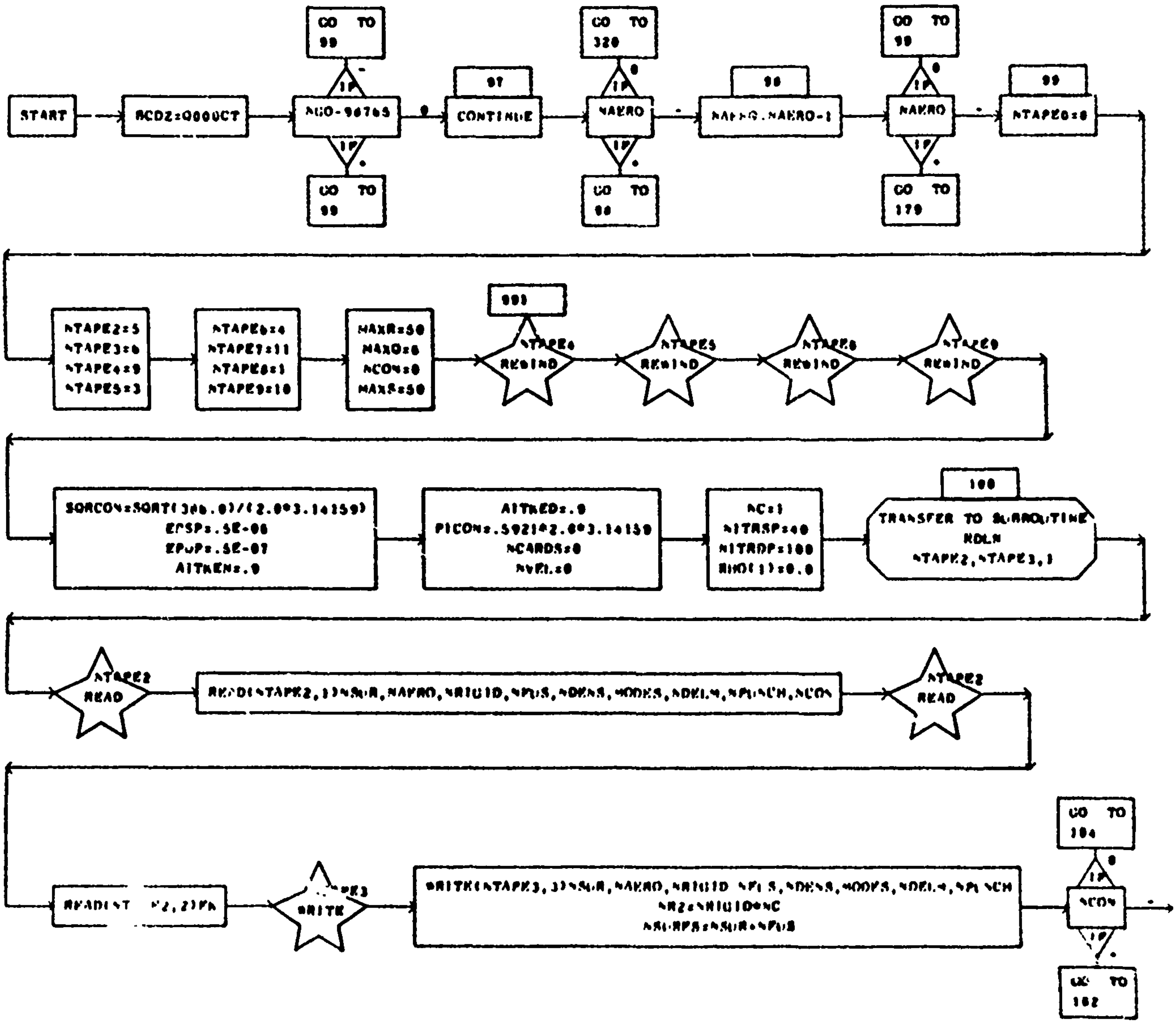
SUBROUTINE MPUNCH(A,M,N,IOUT,ITRA,IORG,BCDZ,MAXM,NTAPE,NCARDS)   PAGE 1

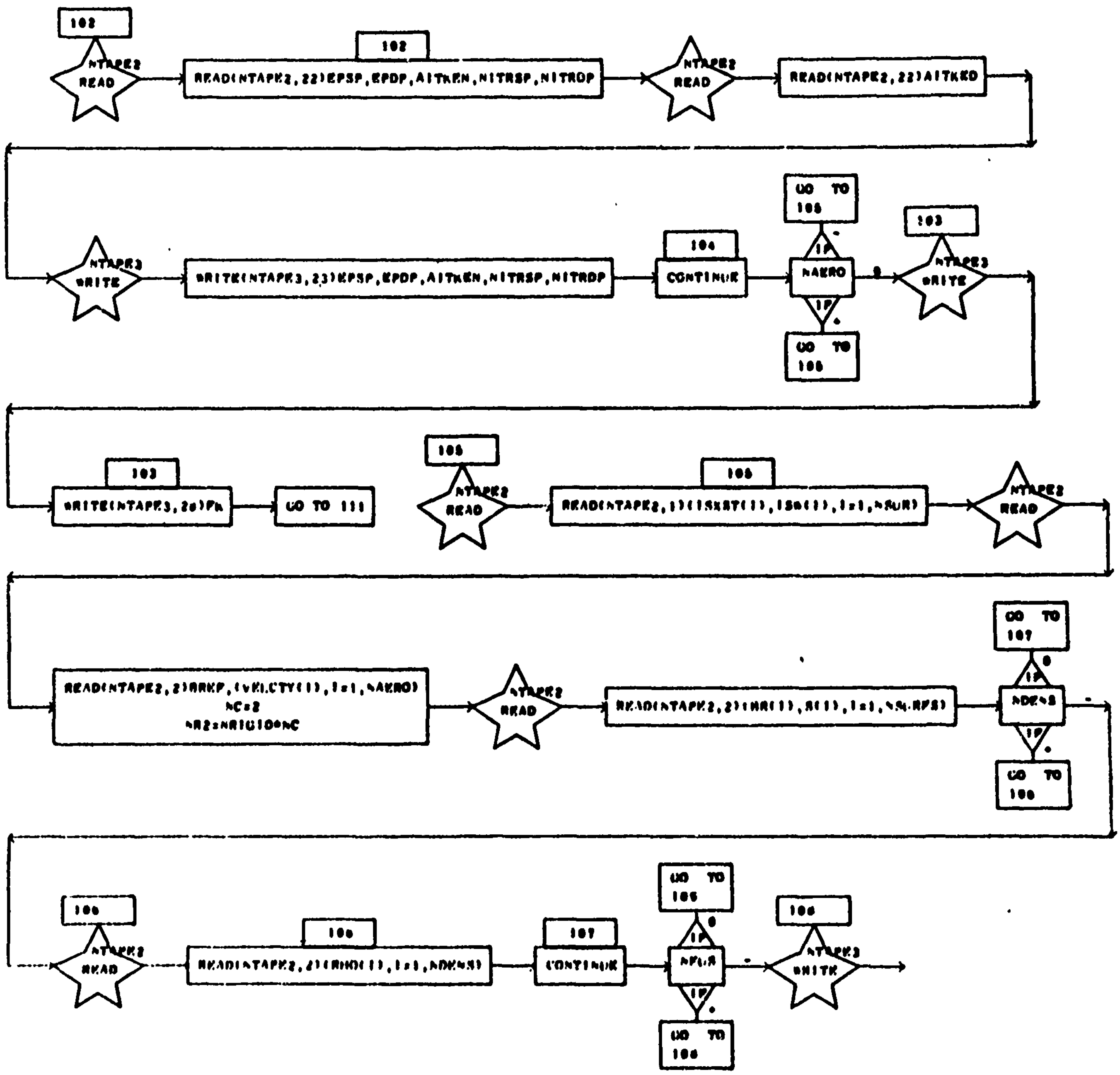


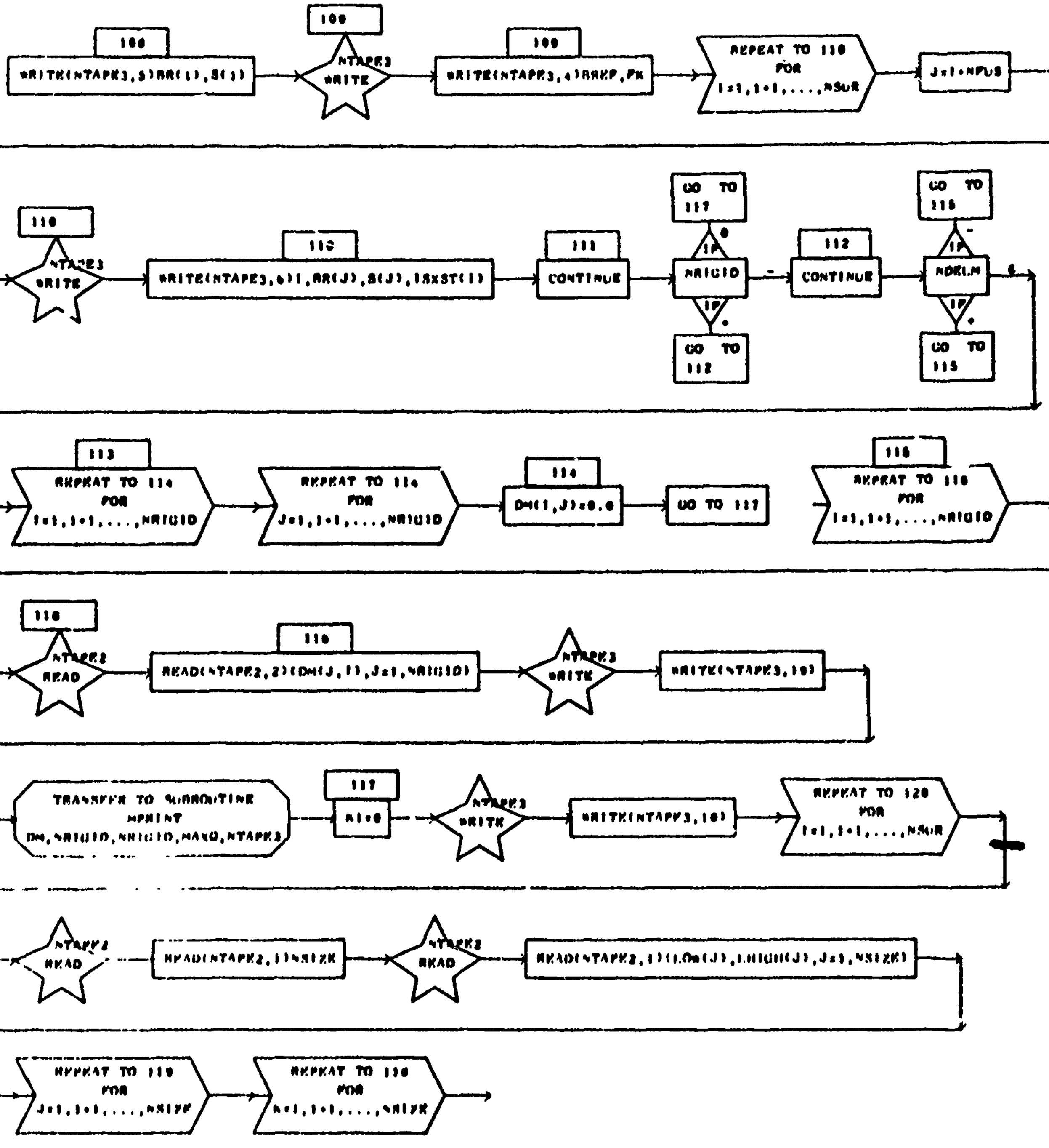
PART1

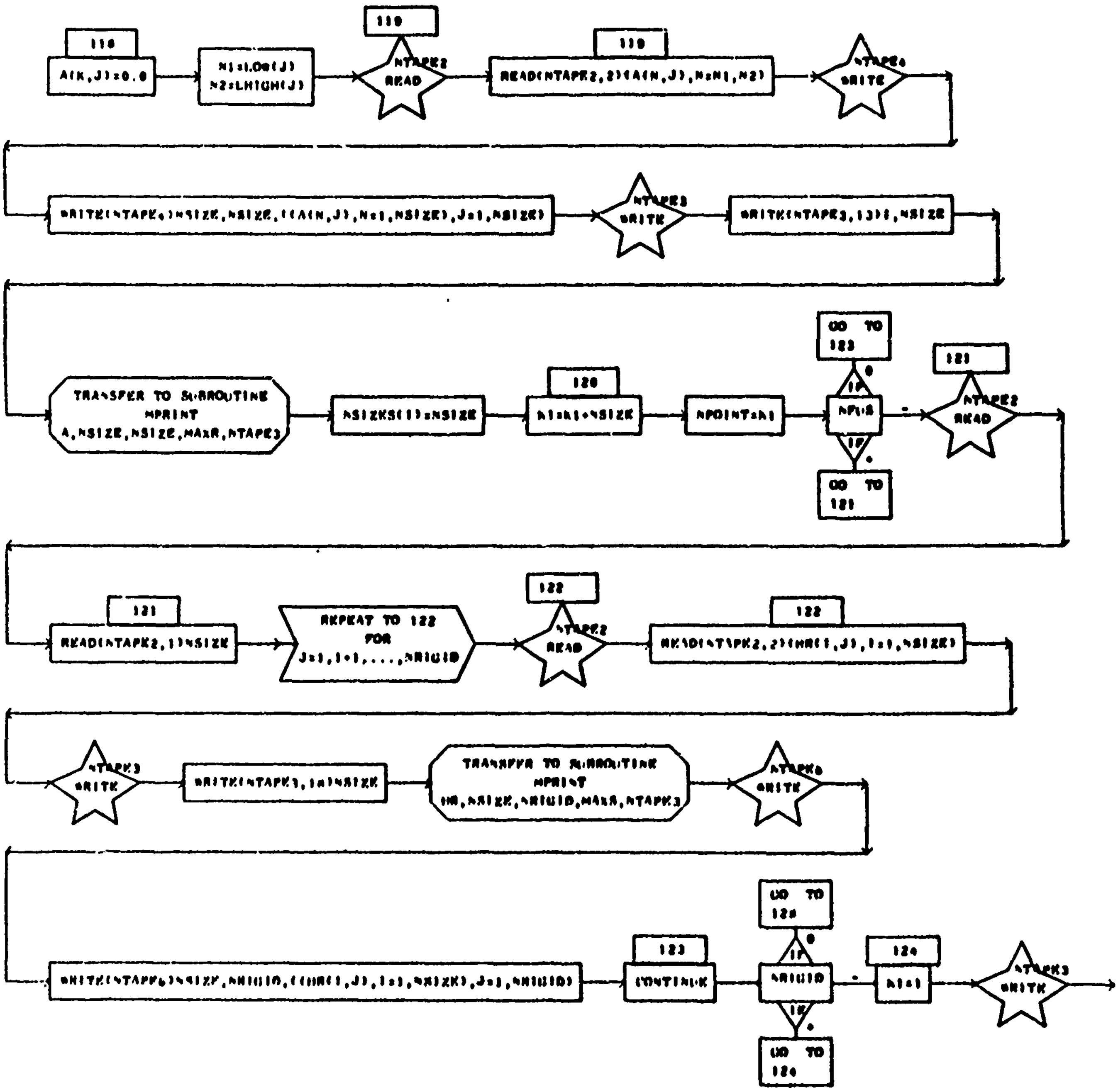
D I M E N S I O N E D   V A R I A B L E S

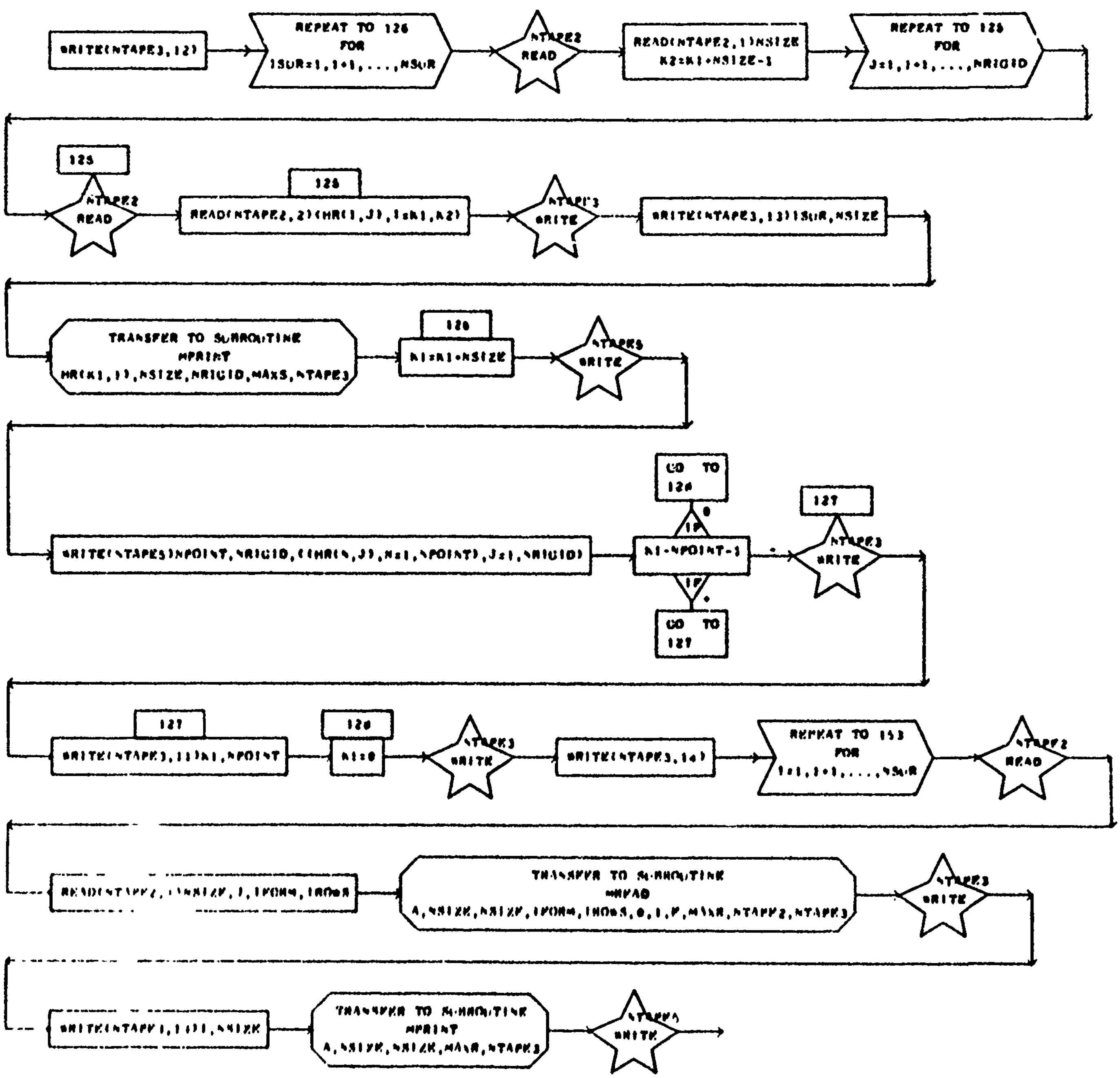
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
ISXST	20	ISO	20	BR	21	S	21	RHO	20
DM	6,6	DQ	6,12	DMBAR	6,12	RARMRR	6,12	LOW	50
LHIGH	50	IT	210	VELCTY	20	NSIZES	20	A	50,100
P	50,100	U	6,100	MR	50,6	MRT	6,100	MY	6,100
Q	6,100								

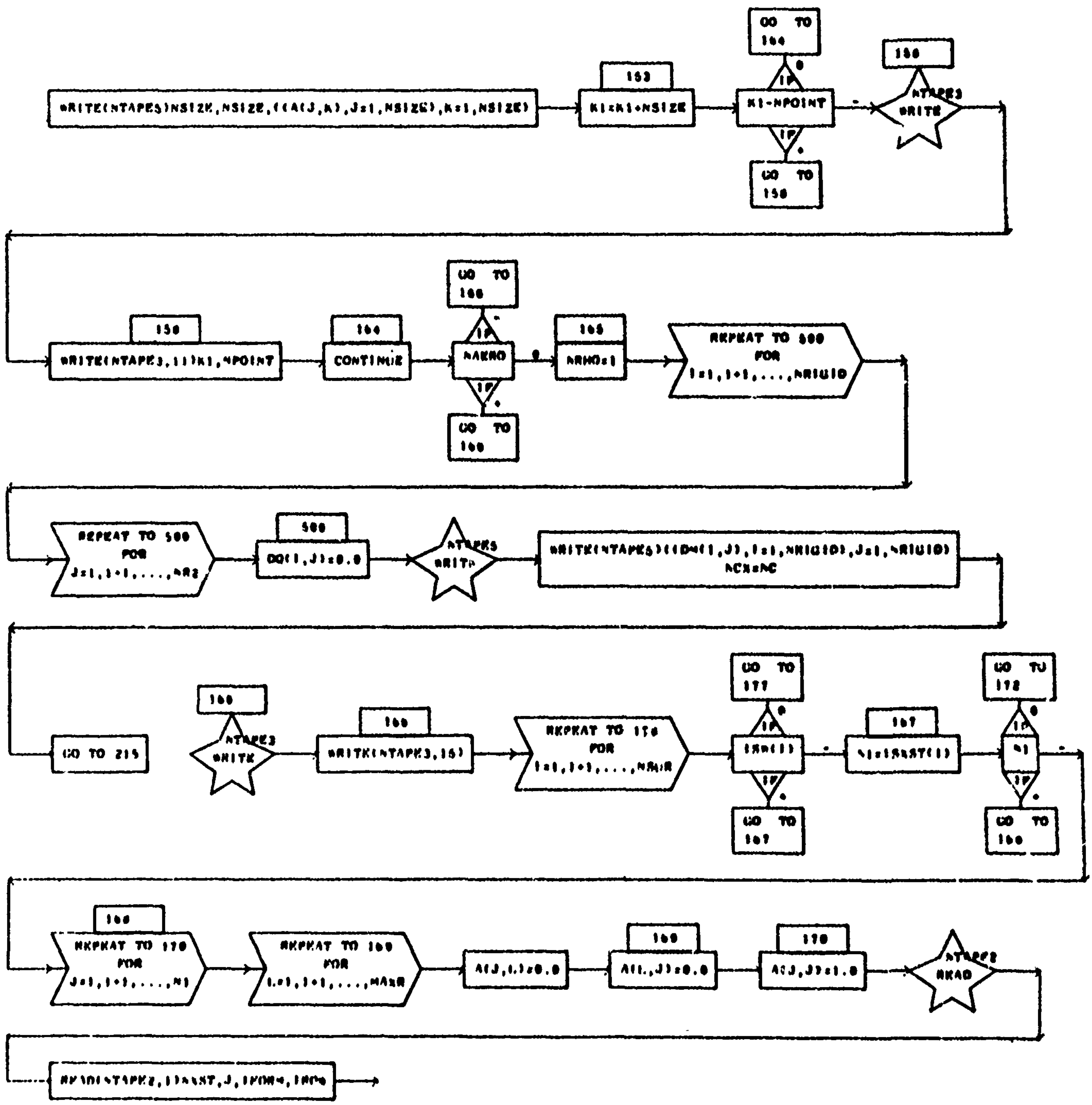

START
GO TO
99
IF
CONTINUE
NAERO
320
NTAPE2=5
NTAPE3=6
NTAPE4=9
NTAPE5=3
NTAPE6=4
NTAPE7=11
NTAPE8=1
NTAPE9=10
MAXR=50
MAXO=6
NCON=0
MAXS=50
REWIND
EPSP=.5E-06
EPUP=.5E-07
AITKEN=.9
AITKED=.9
NCARDS=0
NC=1
NITRSP=40
NITRDP=100
100
TRANSFER TO SUBROUTINE
RDLN
NTAPE2,NTAPE3,1
READ
WRITE
NCON
GO TO
104
102

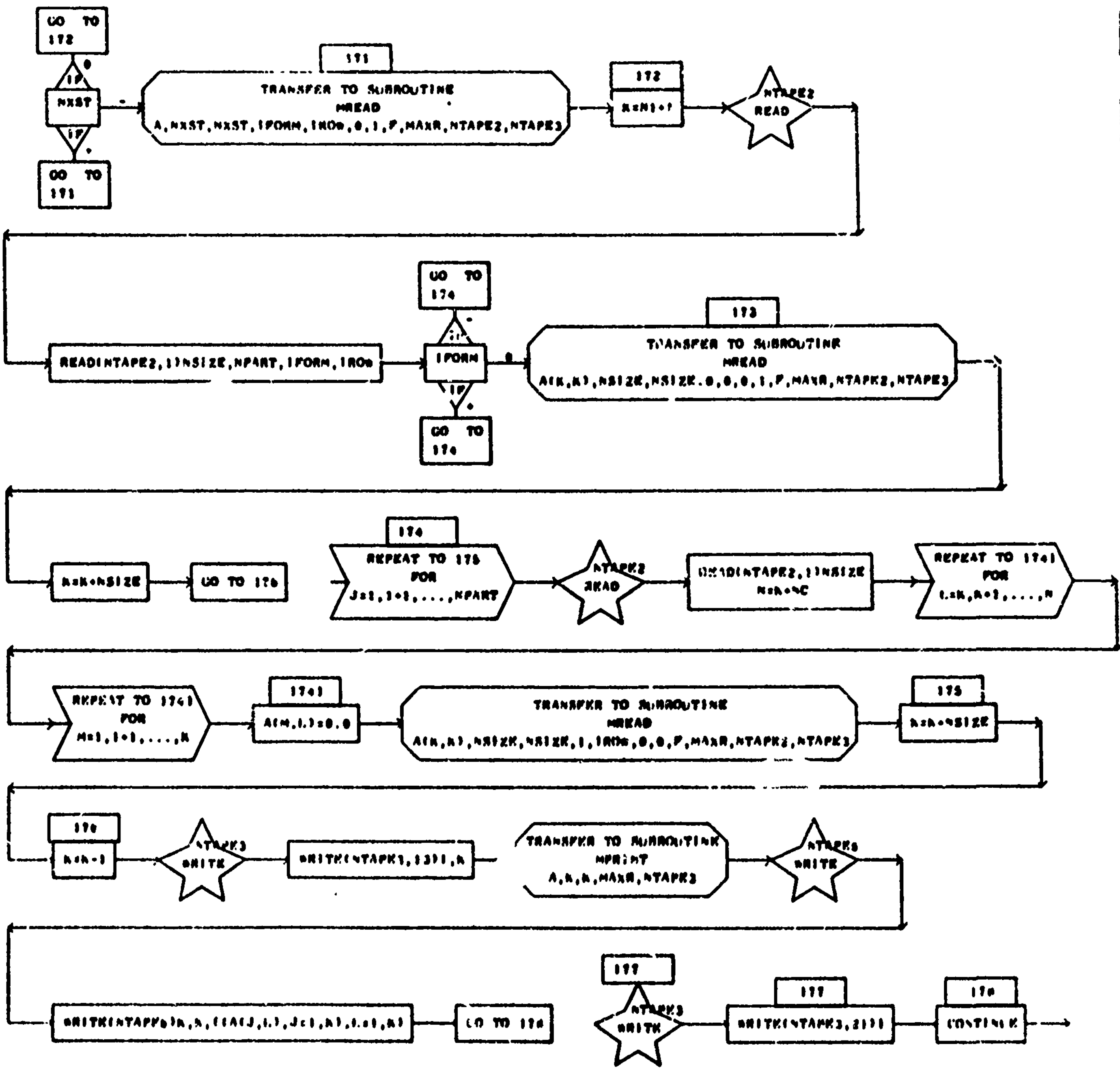

102
NTAPE2
READ
READ(NTAPE2,22)EPSP,EPDP,AITKEN,NITRSP,NITRDP
READ(NTAPE2,22)AITKED
NTAPE3
WRITE
CONTINUE
NAERO
GO TO

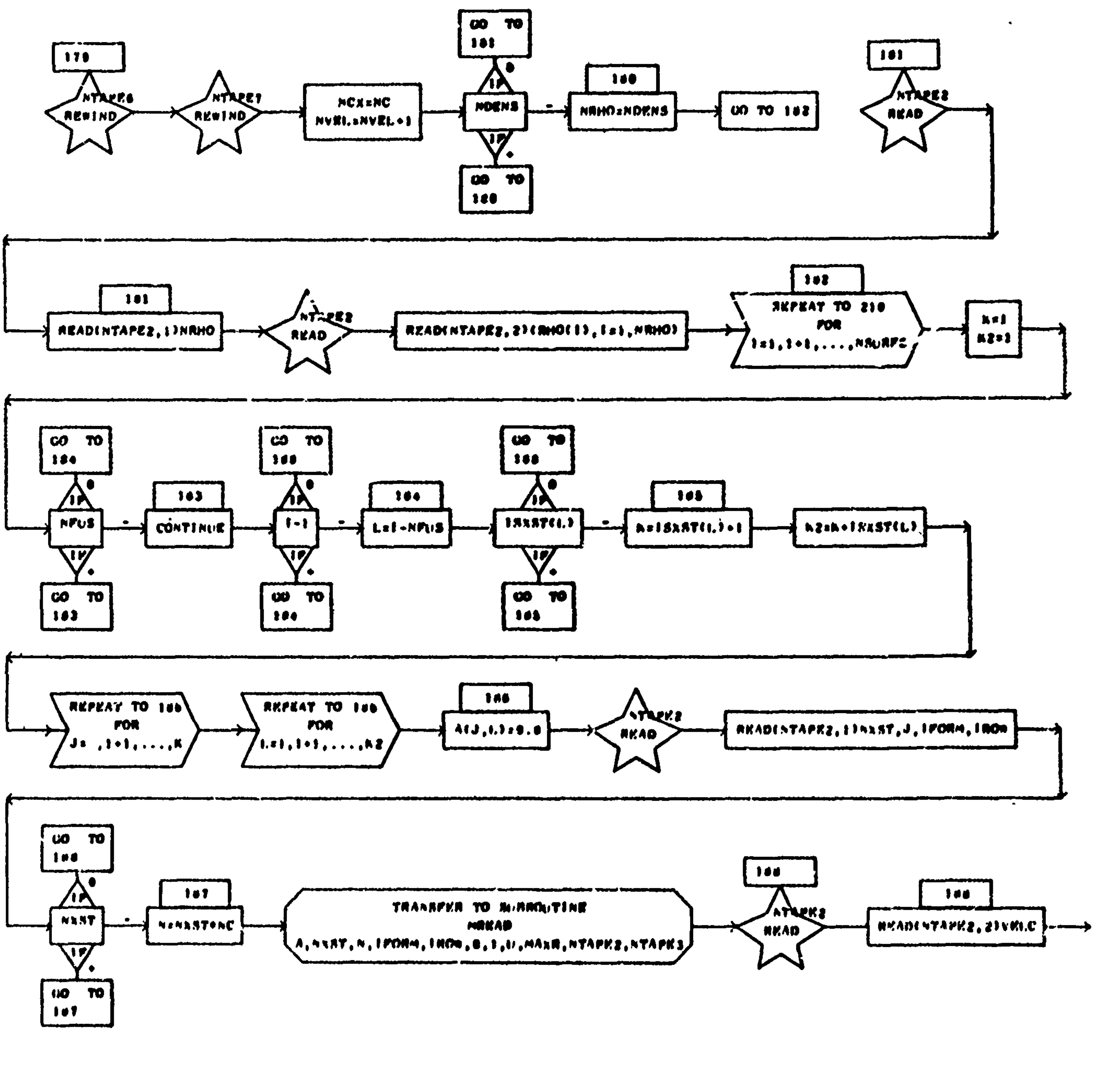



118
A(K,J)=0.0
N1=LOW(J)
N2=LHIGH(J)
119
NTAPE2
READ
119
READ(NTAPE2,2)(A(N,J),N=N1,N2)
NTAPE4
WRITE
WRITE(NTAPE4)NSIZE,NSIZE,((A(N,J),N=1,NSIZE),J=1,NSIZE)
NTAPE3
WRITE
WRITE(NTAPE3,13)I,NSIZE
TRANSFER TO SUBROUTINE
MPRINT
A,NSIZE,NSIZE,MAXR,NTAPE3
NSIZES(I)=NSIZE
120
N1=N1+NSIZE
NPOINT=N1
GO TO
123
IF
NFUS
IF
GO TO
121
121
NTAPE2
READ
121
READ(NTAPE2,1)NSIZE
REPEAT TO 122
FOR
J=1,1+1,...,NRIGID
122
NTAPE2
READ
122
NTAPE3
WRITE
TRANSFER TO SUBROUTINE
MPRINT
NTAPE4
WRITE
123
CONTINUE
GO TO
IF
NRIGID
IF
GO TO
NTAPE3
WRITE

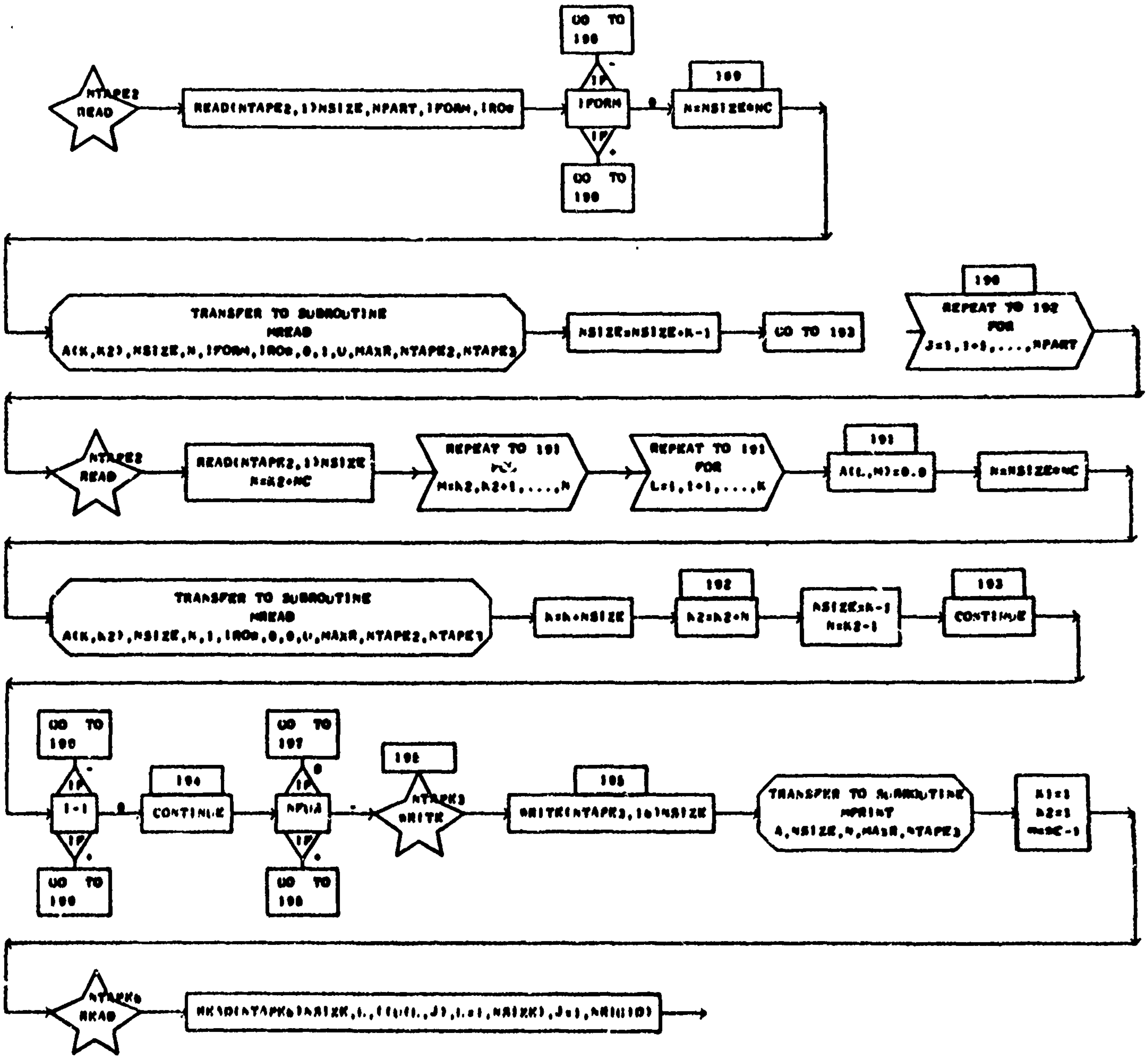

WRITE(NTAPE3,12)
REPEAT TO 126
FOR
ISUR=1,1+1,...,NSUR
NTAPE2
READ
READ(NTAPE2,1)NSIZE
K2=K1+NSIZE-1
REPEAT TO 125
FOR
J=1,1+1,...,NRIGID
125
NTAPE2
READ
125
READ(NTAPE2,2)(HR(I,J),I=K1,K2)
NTAPE3
WRITE
WRITE(NTAPE3,13)ISUR,NSIZE
TRANSFER TO SUBROUTINE
MPRINT
HR(K1,1),NSIZE,NRIGID,MAXS,NTAPE3
126
K1=K1+NSIZE
NTAPE3
WRITE
WRITE(NTAPE5)NPOINT,NRIGID,((HR(N,J),N=1,NPOINT),J=1,NRIGID)
GO TO
128
IF
K1-NPOINT-1
IF
GO TO
127
127
NTAPE3
WRITE
127
WRITE(NTAPE3,11)K1,NPOINT
128
K1=0
NTAPE3
WRITE
WRITE(NTAPE3,14)
REPEAT TO 153
FOR
I=1,1+1,...,NSUR
NTAPE2
READ
TRANSFER TO SUBROUTINE
MREAD
NTAPE3
WRITE
TRANSFER TO SUBROUTINE
MPRINT
A,NSIZE,NSIZE,MAXR,NTAPE3
WRITE

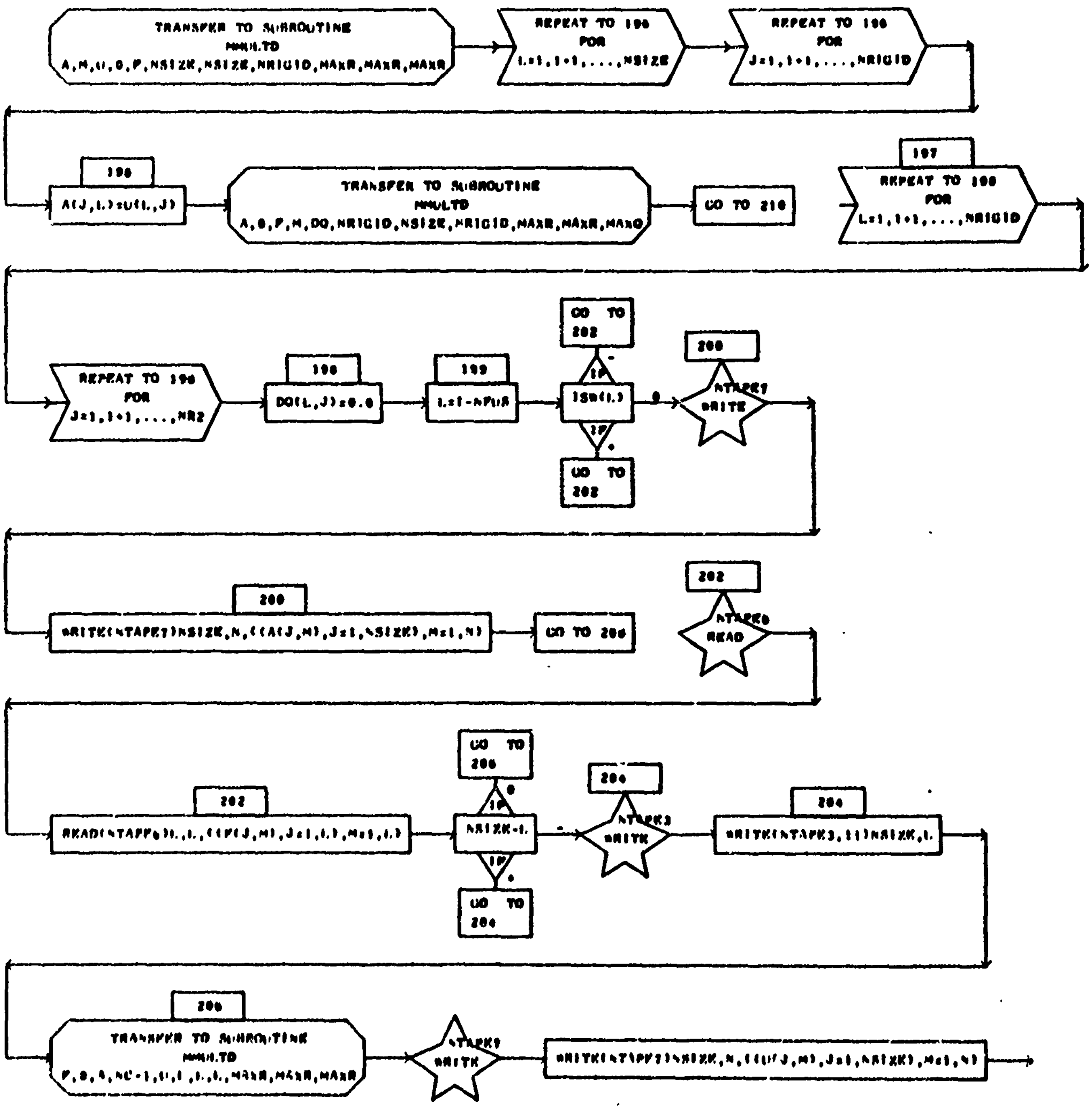

WRITE(NTAPE5)NSIZE,NSIZE,((A(J,K),J=1,NSIZE),K=1,NSIZE)
153
K1=K1+NSIZE
GO TO
K1-NPOINT
GO TO
158
NTAPE3
WRITE
158
WRITE(NTAPE3,1)K1,NPOINT
164
CONTINUE
GO TO
NAERO
GO TO
165
NRHO=1
REPEAT TO 500
FOR
REPEAT TO 500
FOR
500
NTAPE5
WRITE
GO TO 215
NTAPE3
WRITE
REPEAT TO
FOR
GO TO
GO TO
GO TO
GO TO
REPEAT TO 170
FOR
REPEAT TO
FOR
170
NTAPE2
READ

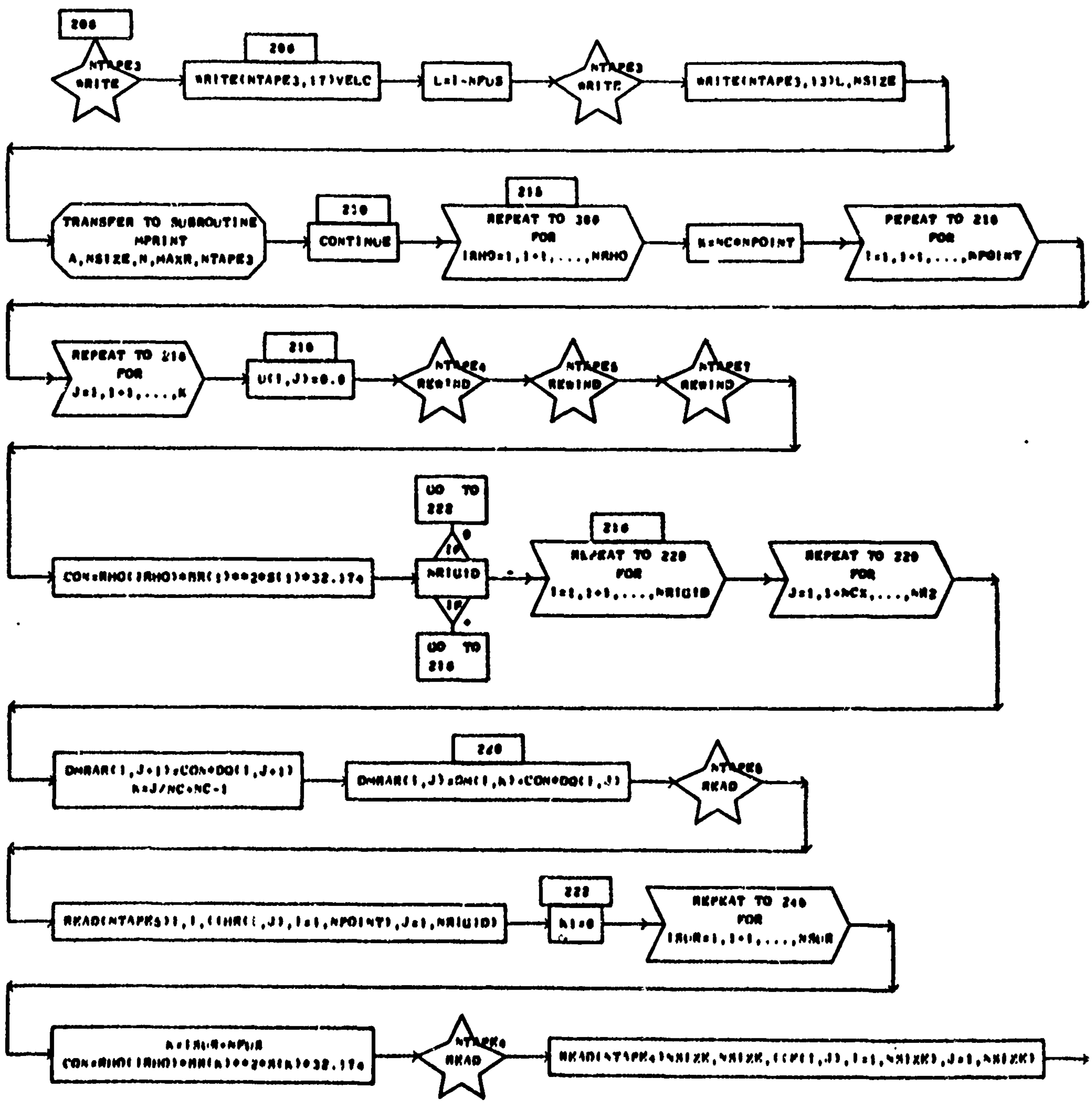

GO TO 172
IF 0
NXST
IF +
GO TO 171
171
TRANSFER TO SUBROUTINE MREAD A,NXST,NXST,IFORM,IROW,0,1,F,MAXR,NTAPE2,NTAPE3
172
N=N1+7
NTAPE2 READ
READ(NTAPE2,1)NSIZE,NPART,IFORM,IROW
GO TO 174
IFORM
GO TO 174
173
TRANSFER TO SUBROUTINE MREAD A(K,K),NSIZE,NSIZE,0,0,0,1,F,MAXR,NTAPE2,NTAPE3
K=K+NSIZE
GO TO 176
174
REPEAT TO 175 FOR J=1,1+1,...,NPART
NTAPE2 READ
READ(NTAPE2,1)NSIZE
REPEAT TO 1741 FOR
REPEAT TO 1741 FOR
1741
A(M,L)=0.0
TRANSFER TO SUBROUTINE MREAD A(K,K),NSIZE,NSIZE,1,IROW,0,0,F,MAXR,NTAPE2,NTAPE3
175
K=K+NSIZE
176
NTAPE3 WRITE
WRITE(NTAPE3,131),K
TRANSFER TO SUBROUTINE MPRINT A,K,K,MAXR,NTAPE3
NTAPE3 WRITE
177
NTAPE3 WRITE
177
WRITE(NTAPE3,2)
178
CONTINUE



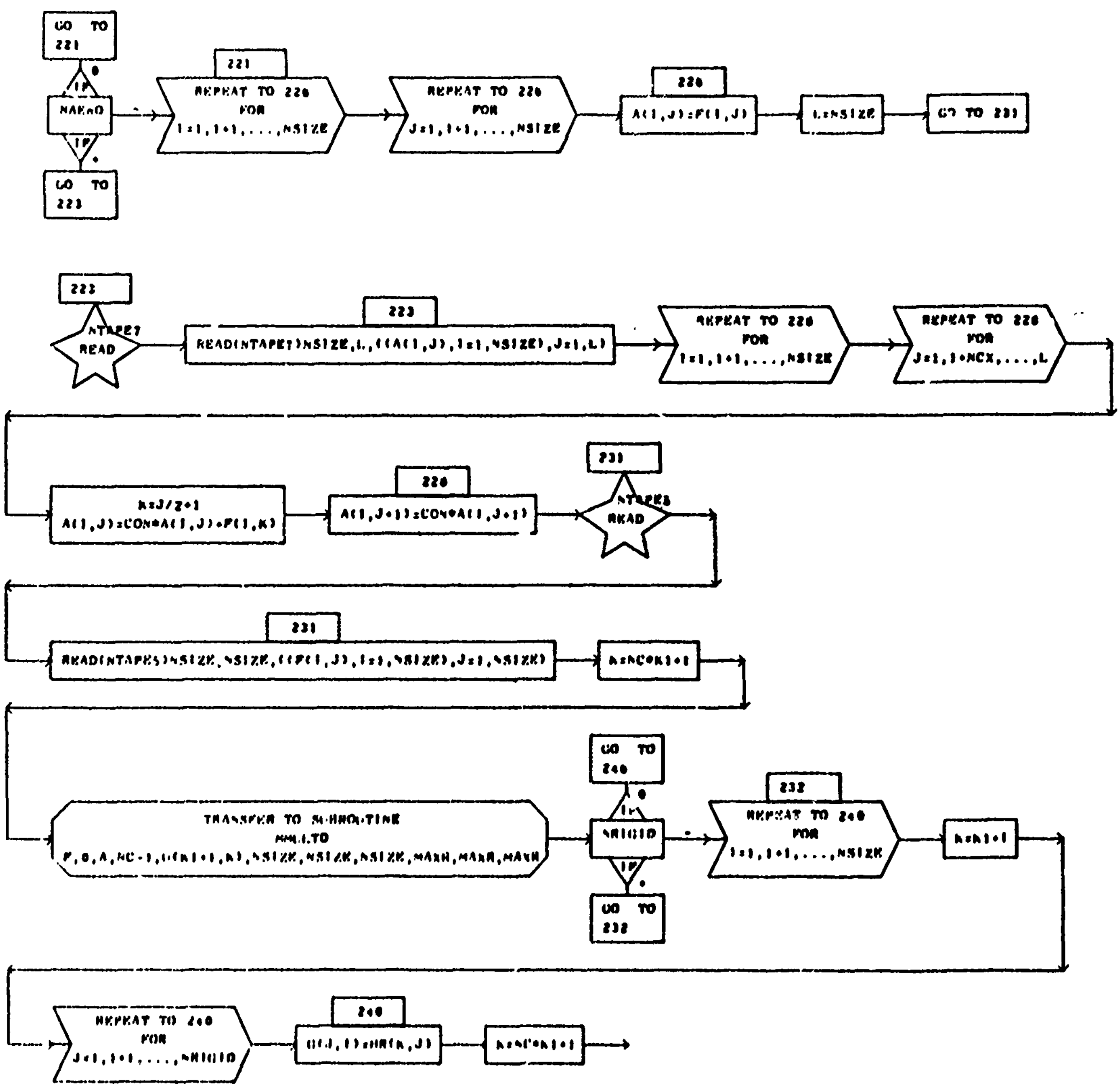
NTAPE6 REWIND
NTAPE1 REWIND
NCX=NC
NVEL=NVEL+1
NDENS
GO TO 180
NRHO=NDENS
GO TO 182
NTAPE2 READ
READ(NTAPE2,1)NRHO
NTAPE2 READ
REPEAT TO 210
FOR
CONTINUE
L=I-NFUS
NFUS
GO TO

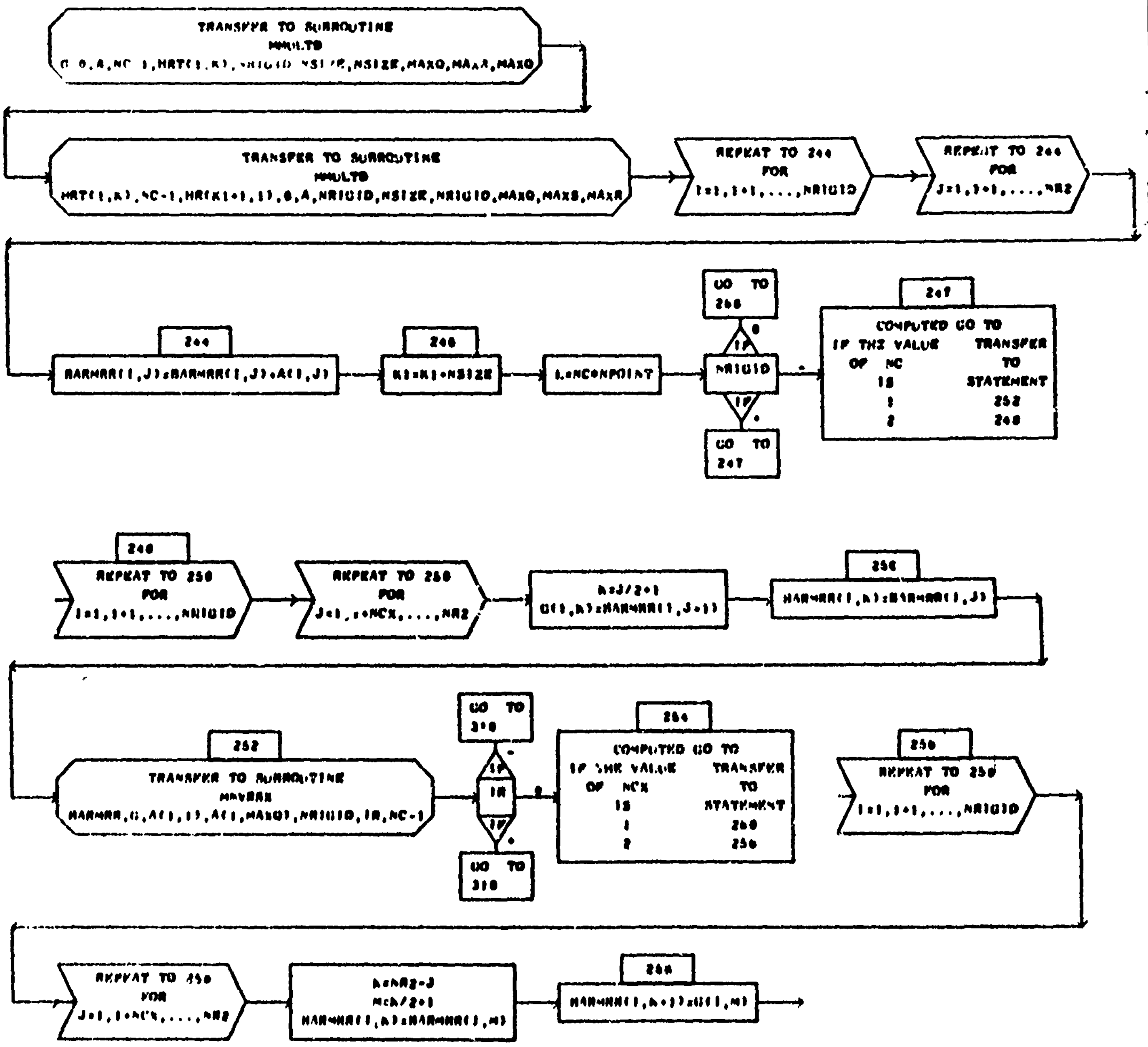

READ
NTAPE2
READ(NTAPE2,1)NSIZE,NPART,IFORM,IROW
IF
IFORM
N=NSIZE*NC
TRANSFER TO SUBROUTINE
MREAD
NSIZE=NSIZE*K-1
REPEAT TO 192
FOR
READ(NTAPE2,1)NSIZE
REPEAT TO 191
FOR
A(L,M)=0.0
N=NSIZE*NC
TRANSFER TO SUBROUTINE
MREAD
CONTINUE
CONTINUE
WRITE
TRANSFER TO SUBROUTINE
MPRINT

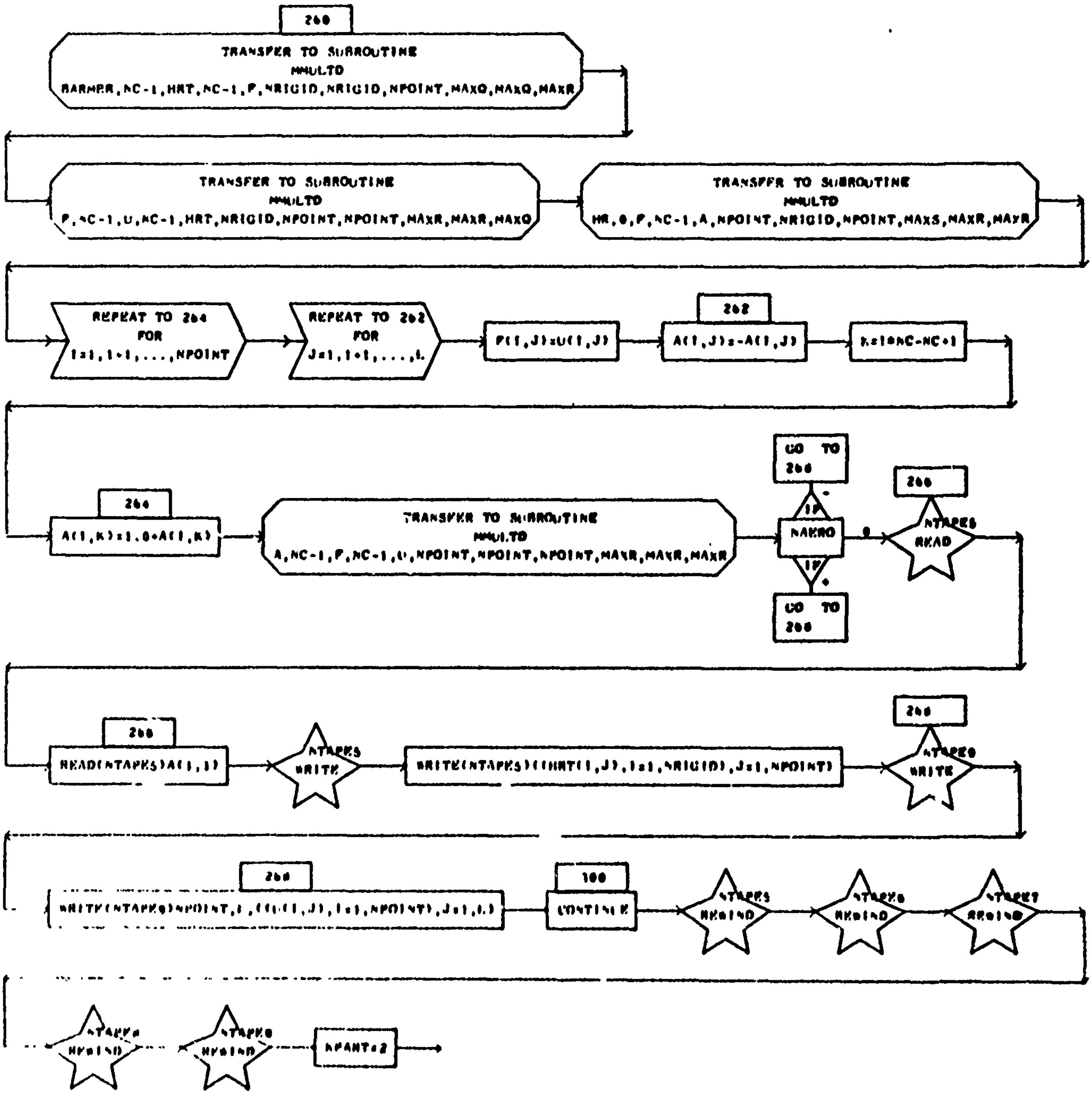

TRANSFER TO SUBROUTINE
REPEAT TO 196
FOR
196
197
REPEAT TO 198
TRANSFER TO SUBROUTINE
MMULTD
GO TO 210
REPEAT TO 198
198
199
GO TO
202
200
WRITE
200
GO TO
202
READ
202
GO TO
206
204
WRITE
204
GO TO
204
206
TRANSFER TO SUBROUTINE
WRITE



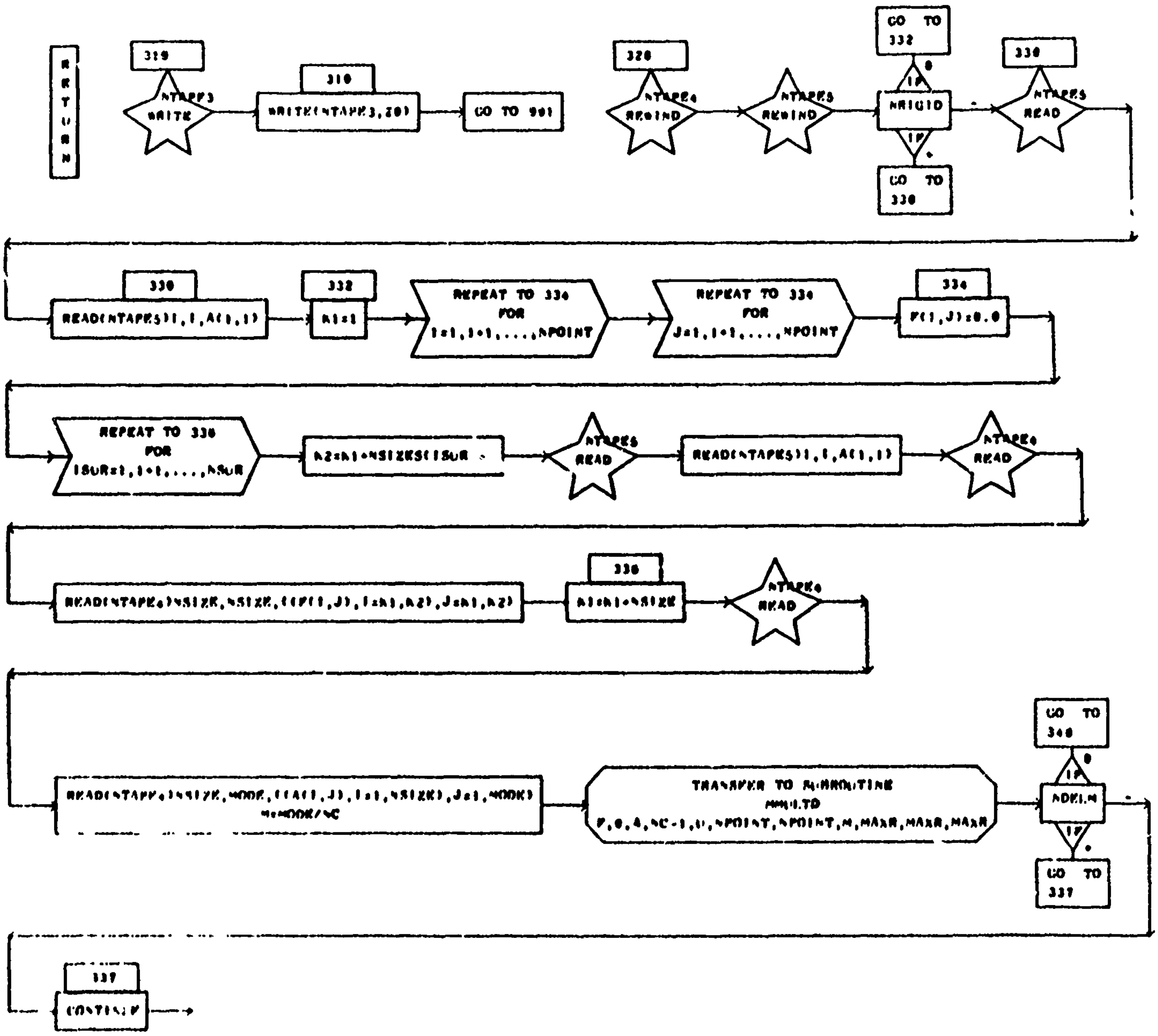
206
NTAPE3 WRITE
208
WRITE(NTAPE3,17)VELC
L=I-NPUS
NTAPE3 WRITE
WRITE(NTAPE3,13)L,NSIZE
TRANSFER TO SUBROUTINE MPRINT A,NSIZE,N,MAXR,NTAPE3
210
CONTINUE
212
REPEAT TO 300 FOR IRHO=1,1+1,...,NRHO
K=NC*NPOINT
REPEAT TO 216 FOR I=1,1+1,...,NPOINT
REPEAT TO 216 FOR J=1,1+1,...,K
216
U(I,J)=0.0
NTAPE4 REWIND
NTAPE6 REWIND
NTAPE7 REWIND
GO TO 222
CON=RHO(IRHO)*RR(I)**2*S(I)*32.174
NRIUID
GO TO 218
218
REPEAT TO 220 FOR I=1,1+1,...,NRIUID
REPEAT TO 220 FOR J=1,1+NC,...,NR2
DMRAR(I,J+1)=CON*DO(I,J+1)
K=J/NC+NC-1
220
DMRAR(I,J)=DM(I,K)+CON*DO(I,J)
NTAPE6 READ
READ(NTAPE5)I,I,((HR(I,J),I=1,NPOINT),J=1,NRIUID)
222
K1=0
REPEAT TO 240 FOR IRUR=1,1+1,...,NRUR
K=IRUR+NPUS
CON=RHO(IRHO)*RR(K)**2*S(K)*32.174
NTAPE4 READ

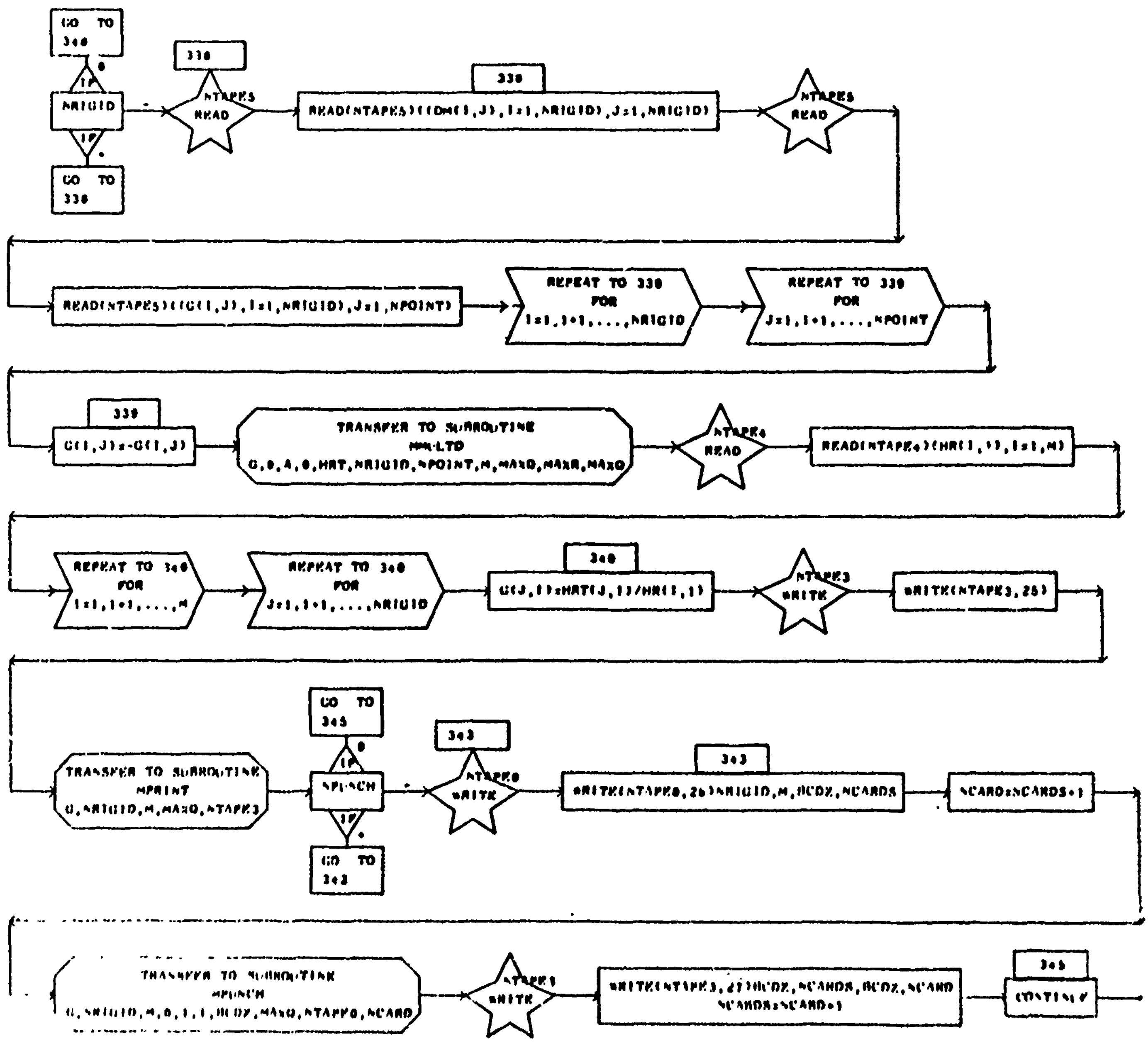

GO TO 221
IF
NAK=0
IF
GO TO 223
221
REPEAT TO 226 FOR I=1,I+1,...,NSIZE
REPEAT TO 226 FOR J=1,I+1,...,NSIZE
226
A(I,J)=F(I,J)
L=NSIZE
GO TO 231
223
NTAPE7 READ
223
READ(NTAPE7)NSIZE,L,((A(I,J),I=1,NSIZE),J=1,L)
REPEAT TO 228 FOR I=1,I+1,...,NSIZE
REPEAT TO 228 FOR J=1,I+NCX,...,L
K=J/2+1
A(I,J)=CON*A(I,J)+F(I,K)
228
A(I,J+1)=CON*A(I,J+1)
231
NTAPE5 READ
231
READ(NTAPE5)NSIZE,NSIZE,((F(I,J),I=1,NSIZE),J=1,NSIZE)
GO TO 240
232
NRIGID
GO TO 232
REPEAT TO 240 FOR I=1,I+1,...,NSIZE
REPEAT TO 240 FOR J=1,I+1,...,NRIGID
240

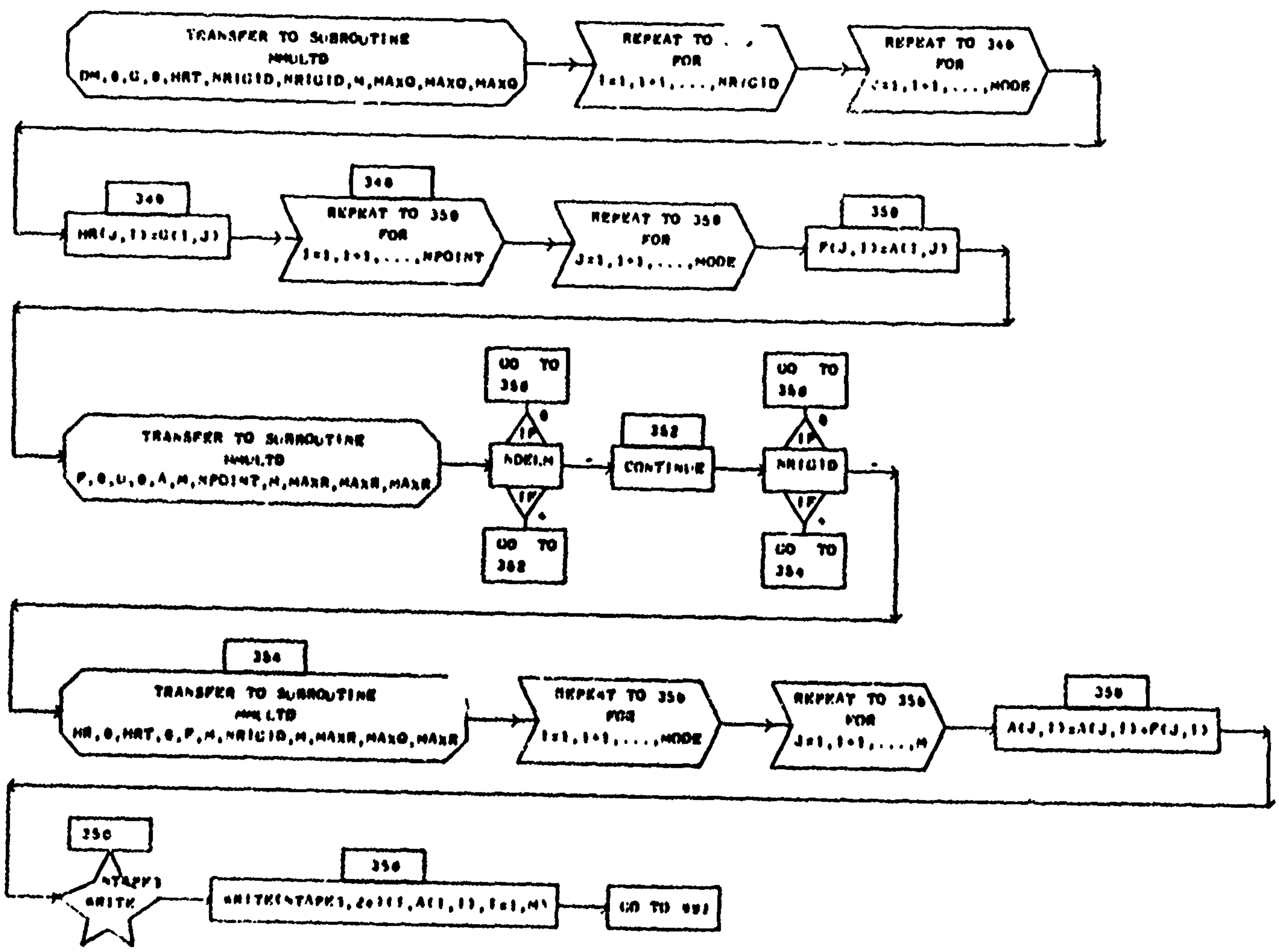

TRANSFER TO SUBROUTINE
MMULTD
TRANSFER TO SUBROUTINE
MMULTD
REPEAT TO 244
FOR
REPEAT TO 244
FOR
244
246
KI=KI+NSIZE
GO TO
GO TO
247
COMPUTED GO TO
IF THE VALUE
OF NC
IS
1
2
TRANSFER
TO
STATEMENT
252
248
248
REPEAT TO 250
FOR
REPEAT TO 250
FOR
252
TRANSFER TO SUBROUTINE
GO TO
GO TO
IR
254
COMPUTED GO TO
TRANSFER
TO
STATEMENT
FOR
FOR



260
TRANSFER TO SUBROUTINE
MMULTD
TRANSFER TO SUBROUTINE
MMULTD
TRANSFER TO SUBROUTINE
MMULTD
REPEAT TO 264
FOR
REPEAT TO 262
FOR
262
264
TRANSFER TO SUBROUTINE
MMULTD
GO TO
IF
NAKRO
IF
GO TO
NTAPES
READ
WRITE
CONTINUE
REWIND


RETURN
319
NTAPE3 WRITE
318
WRITE(NTAPE3,20)
GO TO 901
328
NTAPE4 REWIND
NTAPE5 REWIND
GO TO 332
NRIGID
GO TO 330
330
NTAPE5 READ
330
READ(NTAPE5)I,I,A(I,I)
332
K1=1
REPEAT TO 334 FOR I=1,1+1,...,NPOINT
REPEAT TO 334 FOR J=1,1+1,...,NPOINT
334
F(I,J)=0.0
REPEAT TO 336 FOR ISUR=1,1+1,...,NSUR
K2=K1+NSIZES(ISUR
NTAPE5 READ
READ(NTAPE5)I,I,A(I,I)
NTAPE4 READ
READ(NTAPE4)NSIZE,NSIZE,((F(I,J),I=K1,K2),J=K1,K2)
336
K1=K1+NSIZE
NTAPE4 READ
READ(NTAPE4)NSIZE,MODE,((A(I,J),I=1,NSIZE),J=1,MODE)
M=MODE/NC
TRANSFER TO SUBROUTINE MMULTD
F,0,A,NC-1,U,NPOINT,NPOINT,M,MAXR,MAXR,MAXR
GO TO 340
NDELM
GO TO 337
337
CONTINUE

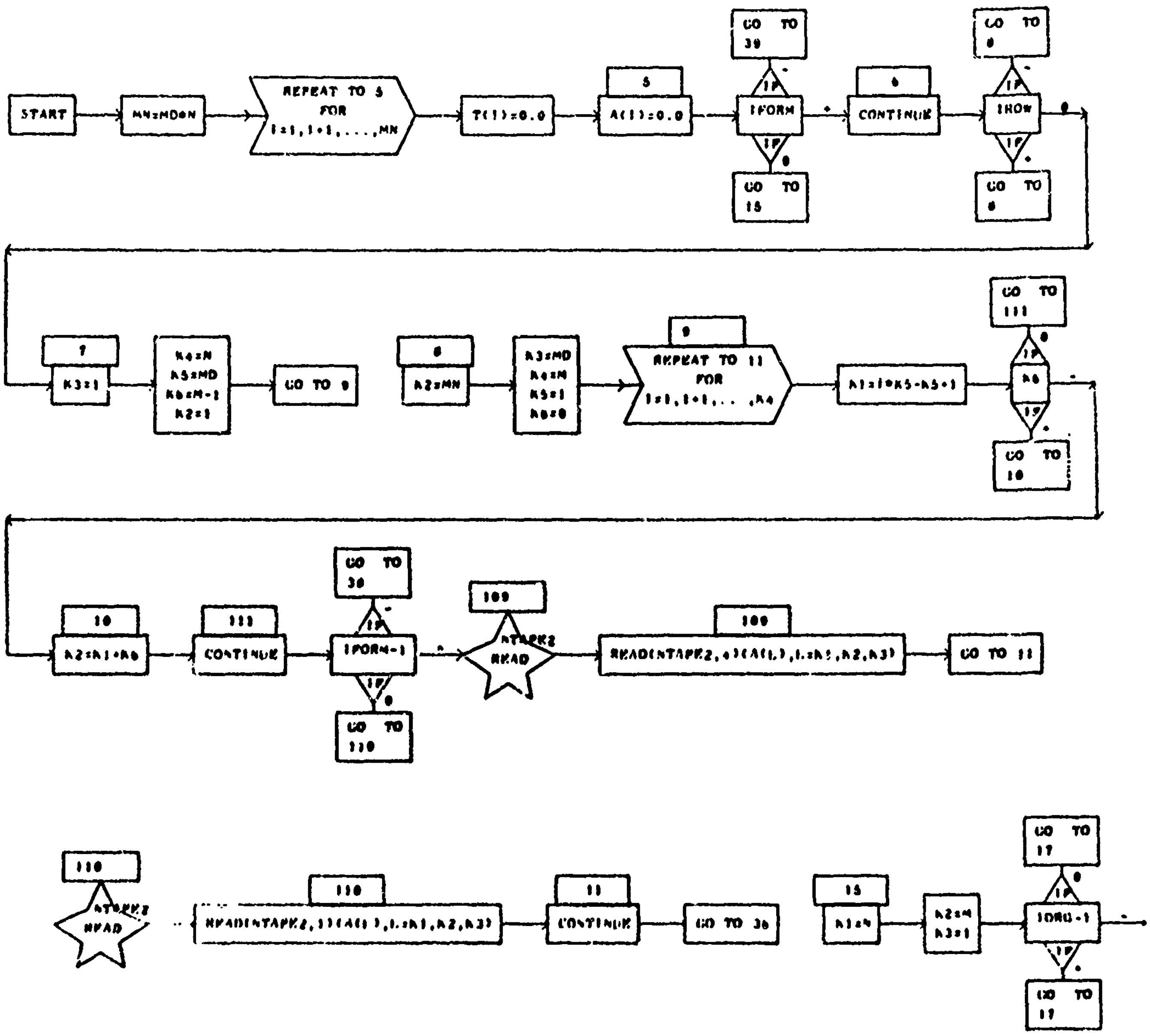

GO TO 340
0
IF
NRIGID
IF
+
GO TO 336
336
NTAPE5 READ
338
READ(NTAPE5)((DH(I,J),I=1,NRIGID),J=1,NRIGID)
NTAPE5 READ
READ(NTAPE5)((G(I,J),I=1,NRIGID),J=1,NPOINT)
REPEAT TO 339 FOR I=1,I+1,...,NRIGID
REPEAT TO 339 FOR J=1,I+1,...,NPOINT
339
G(I,J)=-G(I,J)
TRANSFER TO SUBROUTINE MMLTD G,0,4,0,HRT,NRIGID,NPOINT,M,MAXD,MAXR,MAXO
NTAPE4 READ
READ(NTAPE4)(HR(I,1),I=1,M)
REPEAT TO 340 FOR I=1,I+1,...,M
REPEAT TO 340 FOR J=1,I+1,...,NRIGID
340
G(J,I)=HRT(J,I)/HR(I,1)
NTAPE3 WRITE
WRITE(NTAPE3,25)
TRANSFER TO SUBROUTINE MPRINT G,NRIGID,M,MAXO,NTAPE3
GO TO 345
0
IF
NPUNCH
IF
+
GO TO 343
343
NTAPE9 WRITE
343
WRITE(NTAPE9,26)NRIGID,M,HCDZ,NCARDS
NCARD=NCARDS+1
TRANSFER TO SUBROUTINE MPUNCH G,NRIGID,M,0,1,1,HCDZ,MAXO,NTAPE9,NCARD
NTAPE3 WRITE
WRITE(NTAPE3,27)HCDZ,NCARDS,HCDZ,NCARD
NCARDS=NCARD+1
345
CONTINUE

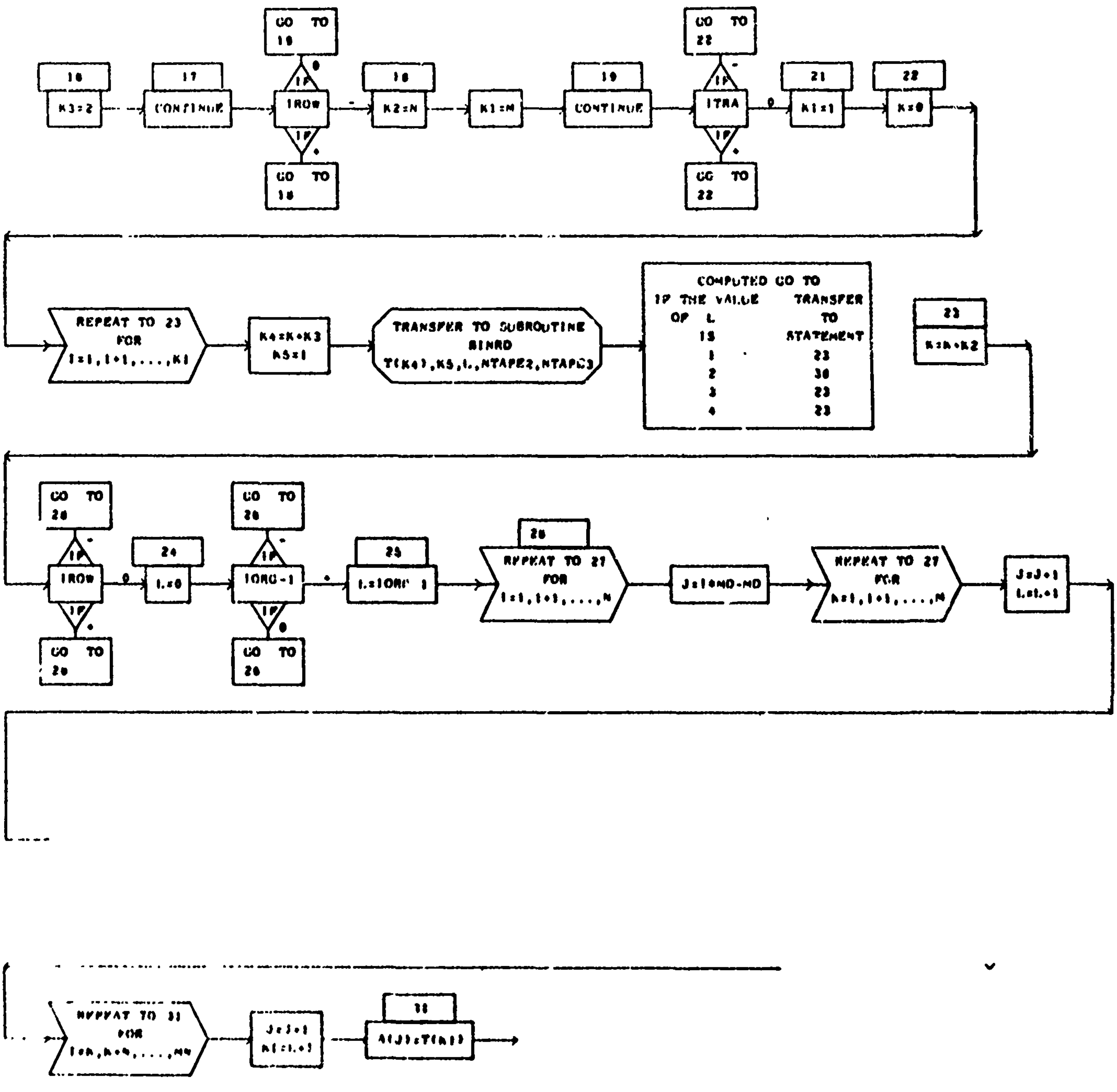

TRANSFER TO SUBROUTINE
MMULTD
DM,0,G,0,HRT,NRIGID,NRIGID,M,MAXO,MAXO,MAXO
REPEAT TO
FOR
I=1,1+1,...,NRIGID
REPEAT TO 340
FOR
J=1,1+1,...,MODE
340
HR(J,I)=G(I,J)
340
REPEAT TO 350
FOR
I=1,1+1,...,NPOINT
REPEAT TO 350
FOR
J=1,1+1,...,MODE
350
F(J,I)=A(I,J)
TRANSFER TO SUBROUTINE
MMULTD
F,0,U,0,A,M,NPOINT,M,MAXR,MAXR,MAXR
GO TO
350
IF
NDELM
IF
GO TO
352
352
CONTINUE
GO TO
350
IF
NRIGID
IF
GO TO
354
354
TRANSFER TO SUBROUTINE
MMULTD
HR,0,HRT,0,F,M,NRIGID,M,MAXR,MAXO,MAXR
REPEAT TO 356
FOR
I=1,1+1,...,MODE
REPEAT TO 356
FOR
J=1,1+1,...,M
356
A(J,I)=A(J,I)+F(J,I)
350
NTAPE1
WRITE
350
WRITE(NTAPE1,2011)(A(I,1),I=1,M)
GO TO 991

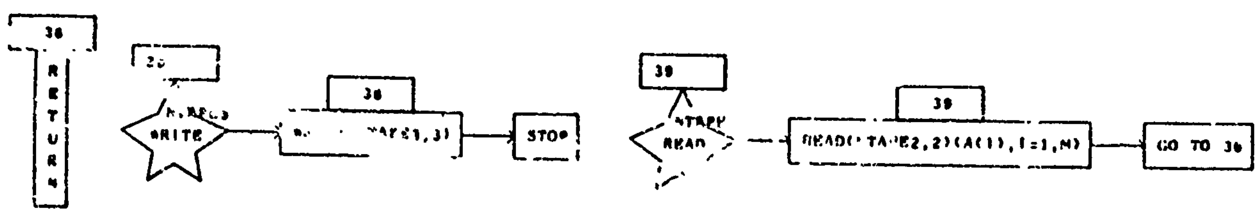
```
                         MREAD

                          MREAD

                  MATRIX READ SUBROUTINE

 CALL MREAD   (A,M,N,IFORM,IROW,ITRA,IORG,T,MD,NTAPE2,NTAPE3     )


      A = MATRIX TO READ IN        ITRA = 0, TRA CARD AFTER MATRIX

     M = NUMBER OF ROWS                 =+1, TRA CARD AFTER EACH ROW

      N = NUMBER OF COLUMNS                          (OR COLUMN )

      IFORM = -1, FORMAT(12A6)    IORG = ORIGIN OF FIRST C.B. CARD

               =  0, COLUMN BINARY    T =MDxN TEMPORARY CELLS

             = +1, FORMAT(5E12.8)  MD= DIMENSIONED NUMBER OF ROWS

      IROW = .0, MATRIX BY COLUMNS                              IN A

                = +1, MATRIX BY ROWS   NTAPE2 = INPUT TAPE

                                       NTAPE3 = OUTPUT TAPE
```

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	T	1						


START
REPEAT TO 5
FOR
T(I)=0.0
A(I)=0.0
IFORM
CONTINUE
IROW
GO TO
K3=1
GO TO 9
REPEAT TO 11
FOR
CONTINUE
IFORM-1
NTAPE2
READ
GO TO 11
CONTINUE
GO TO 36
IORU-1


16
K3=2
17
CONTINUE
GO TO 19
IF
IROW
IF
GO TO 18
18
K2=N
K1=M
19
CONTINUE
GO TO 22
ITRA
GO TO 22
21
K1=1
22
K=0
REPEAT TO 23 FOR I=1,I+1,...,K1
K4=K+K3
K5=1
TRANSFER TO SUBROUTINE BINRD T(K4),K5,L,NTAPE2,NTAPE3
COMPUTED GO TO
IF THE VALUE OF L IS
TRANSFER TO STATEMENT
1 23
2 30
3 23
4 23
23
K=K+K2
GO TO 24
IROW
GO TO 24
24
L=0
GO TO 26
IORG-1
GO TO 26
25
L=IORG 1
26
REPEAT TO 27 FOR I=1,I+1,...,N
J=I+MD-MD
REPEAT TO 27 FOR K=1,I+1,...,M
J=J+1
L=L+1
REPEAT TO 31 FOR I=K,K+N,...,MN
J=J+1
K1=L+1
31
A(J)=T(K1)

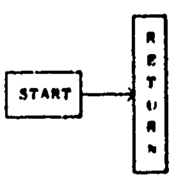

36
RETURN
WRITE
38
STOP
39
NTAPE
READ
39
GO TO 36

BINRD

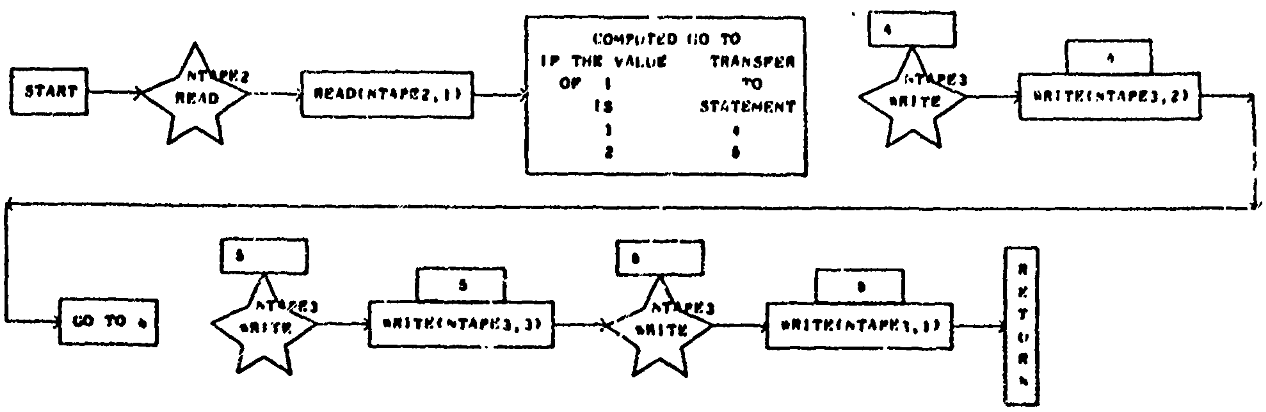
D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
T	1								

SUBROUTINE BINRD (T,K,L,NTAPE1,NTAPE2) PAGE 1



SUBROUTINE RDLN (NTAPE2, NTAPE3, I) PAGE 1

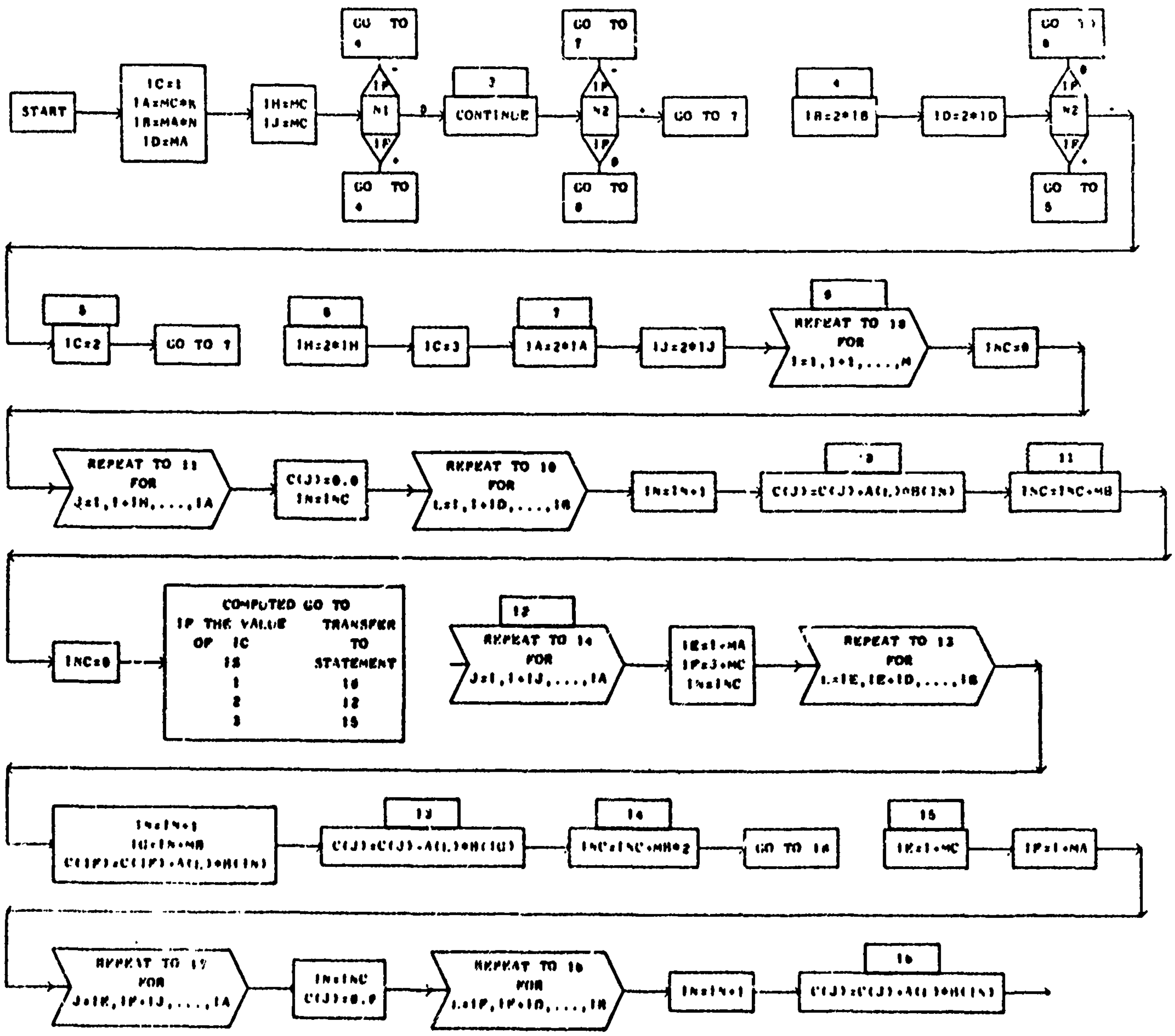


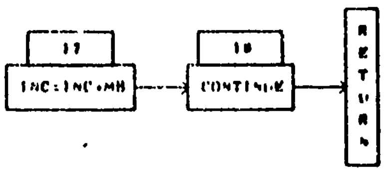
MMULTO

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	B	1	C	1				

SUBROUTINE MMULTD (A,N1,B,N2,C,M,N,K,MA,MB,MC) PAGE 1

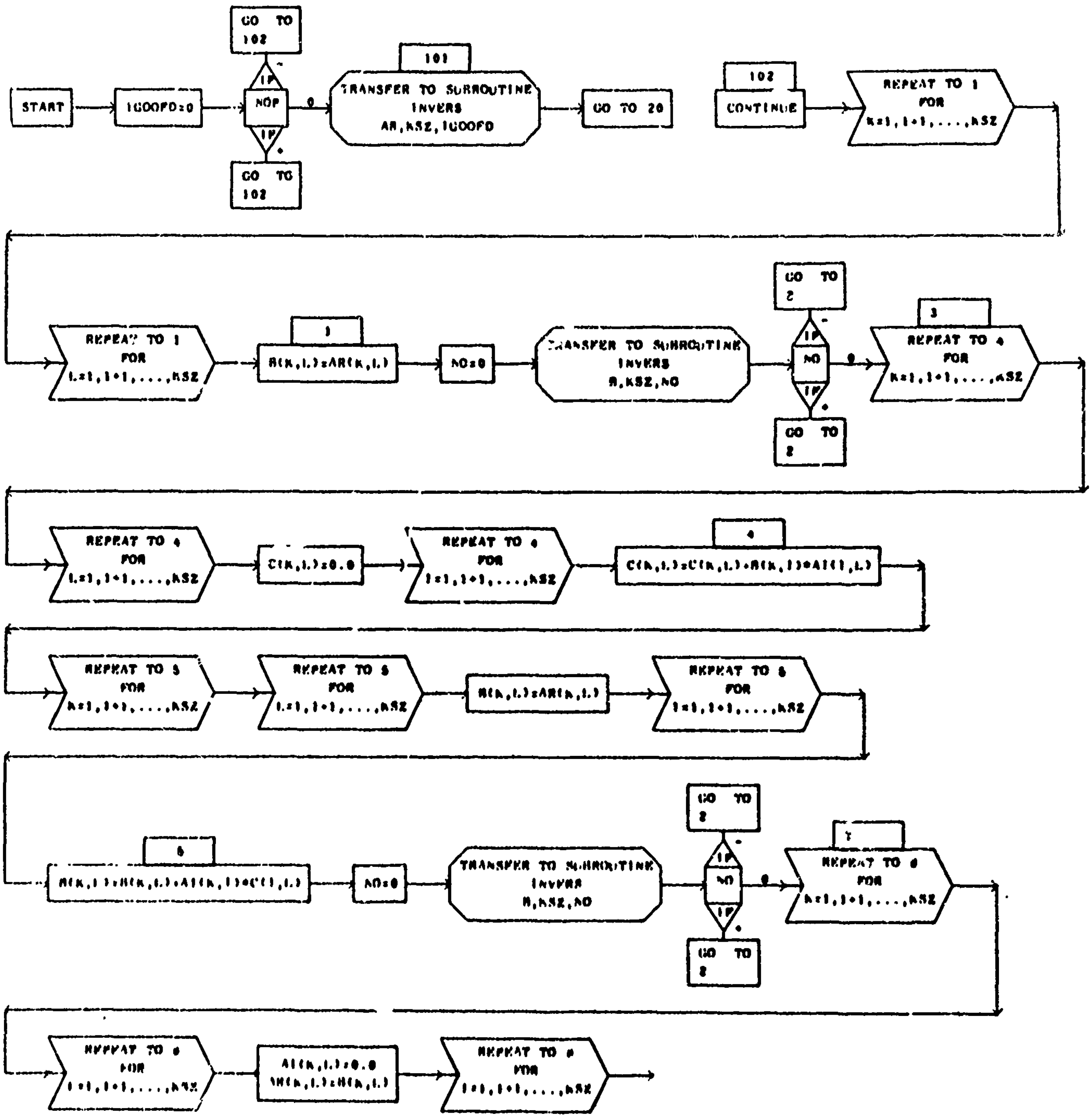


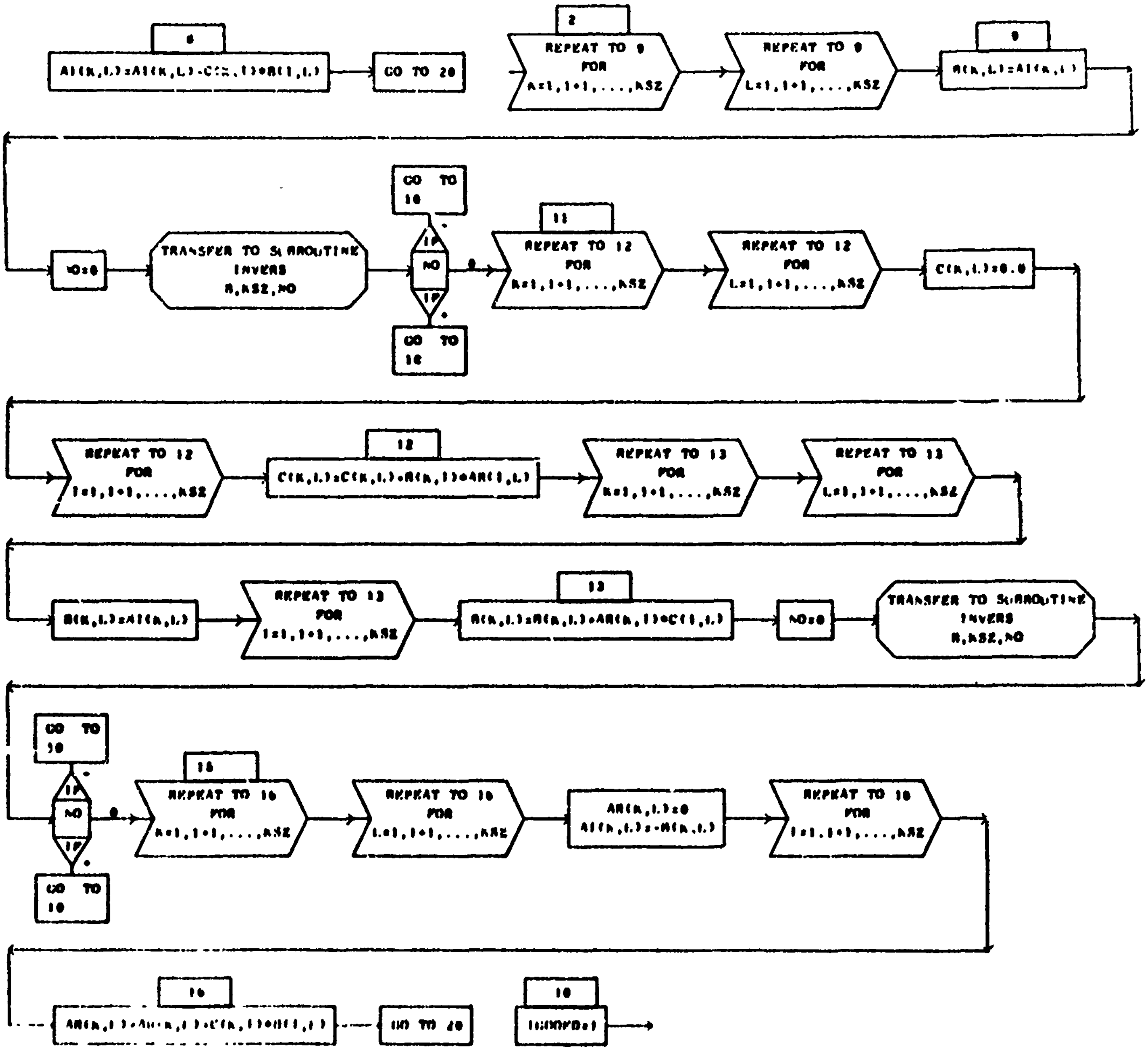

17
INC=INC+MB
18
CONTINUE
RETURN

MNVRSS

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
AR	6,6	AI	6,6	B	6,6	C	6,6		


START
IGOOFD=0
IF NOP
GO TO 102
101
TRANSFER TO SUBROUTINE INVERS AR,KSZ,IGOOFD
GO TO 20
102
CONTINUE
REPEAT TO 1 FOR K=1,1+1,...,KSZ
REPEAT TO 1 FOR L=1,1+1,...,KSZ
1
B(K,L)=AR(K,L)
NO=0
TRANSFER TO SUBROUTINE INVERS B,KSZ,NO
IF NO
GO TO 2
3
REPEAT TO 4 FOR K=1,1+1,...,KSZ
REPEAT TO 4 FOR L=1,1+1,...,KSZ
C(K,L)=0.0
REPEAT TO 4 FOR I=1,1+1,...,KSZ
4
C(K,L)=C(K,L)+B(K,I)*AI(I,L)
REPEAT TO 5 FOR K=1,1+1,...,KSZ
REPEAT TO 5 FOR L=1,1+1,...,KSZ
B(K,L)=AR(K,L)
REPEAT TO 5 FOR I=1,1+1,...,KSZ
5
B(K,L)=B(K,L)+AI(K,I)*C(I,L)
NO=0
TRANSFER TO SUBROUTINE INVERS B,KSZ,NO
IF NO
GO TO 2
7
REPEAT TO 8 FOR K=1,1+1,...,KSZ
REPEAT TO 8 FOR L=1,1+1,...,KSZ
AI(K,L)=0.0
AR(K,L)=B(K,L)
REPEAT TO 8 FOR I=1,1+1,...,KSZ


6
AI(K,L)=AI(K,L)-C(K,I)*B(I,L)
GO TO 20
2
REPEAT TO 9 FOR K=1,1+1,...,KSZ
REPEAT TO 9 FOR L=1,1+1,...,KSZ
9
B(K,L)=AI(K,L)
NO=0
TRANSFER TO SUBROUTINE INVERS B,KSZ,NO
GO TO 10
IF NO
11
REPEAT TO 12 FOR K=1,1+1,...,KSZ
REPEAT TO 12 FOR L=1,1+1,...,KSZ
C(K,L)=0.0
REPEAT TO 12 FOR I=1,1+1,...,KSZ
12
C(K,L)=C(K,L)+B(K,I)*AR(I,L)
REPEAT TO 13 FOR K=1,1+1,...,KSZ
REPEAT TO 13 FOR L=1,1+1,...,KSZ
B(K,L)=AI(K,L)
REPEAT TO 13 FOR I=1,1+1,...,KSZ
13
B(K,L)=B(K,L)+AR(K,I)*C(I,L)
NO=0
TRANSFER TO SUBROUTINE INVERS B,KSZ,NO
GO TO 10
15
REPEAT TO 16 FOR K=1,1+1,...,KSZ
REPEAT TO 16 FOR L=1,1+1,...,KSZ
AR(K,L)=0
AI(K,L)=-B(K,L)
REPEAT TO 16 FOR I=1,1+1,...,KSZ
16
AR(K,L)=AR(K,L)+C(K,I)*B(I,L)
GO TO 20
10
IGOOFD=1

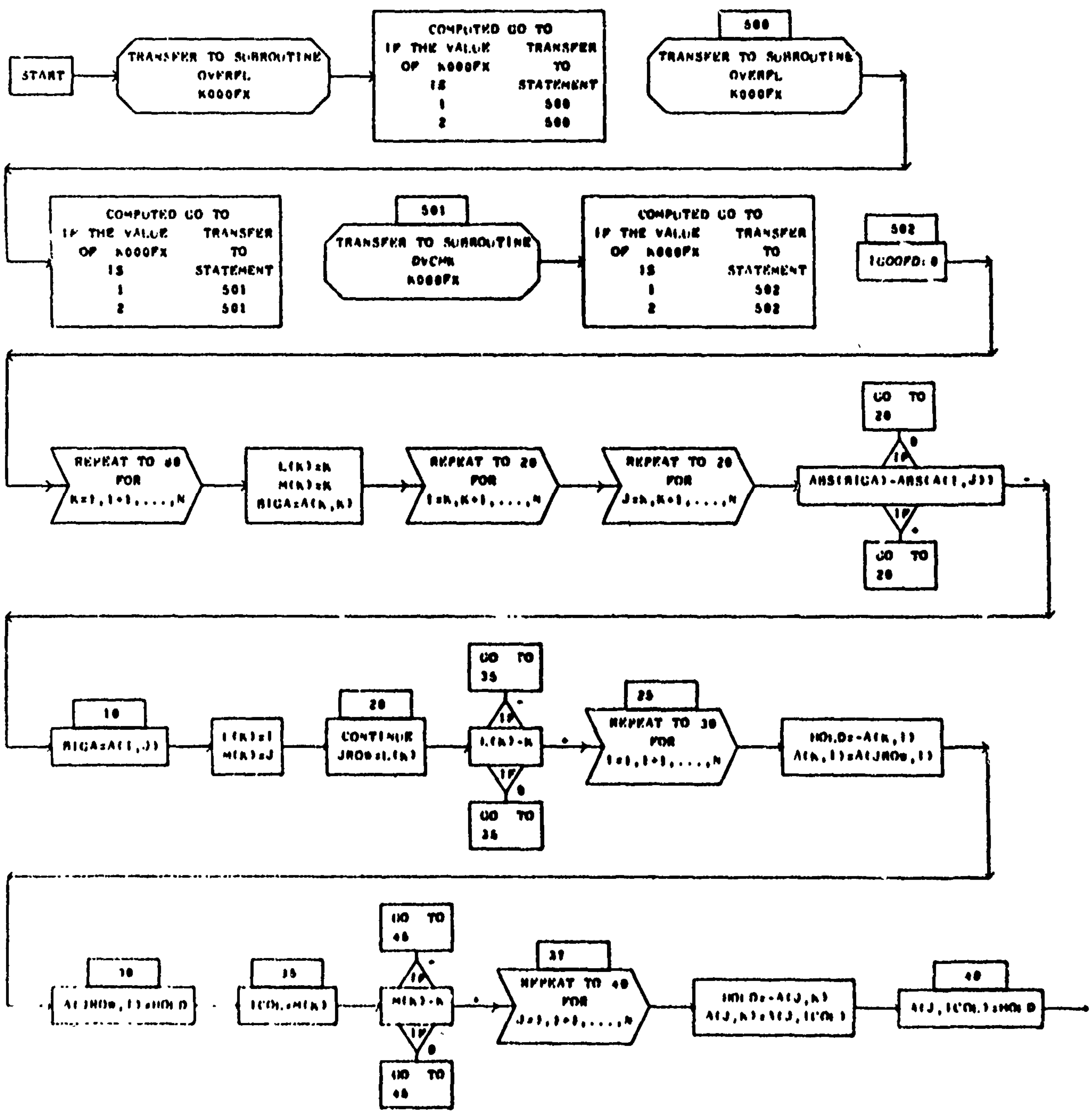
20

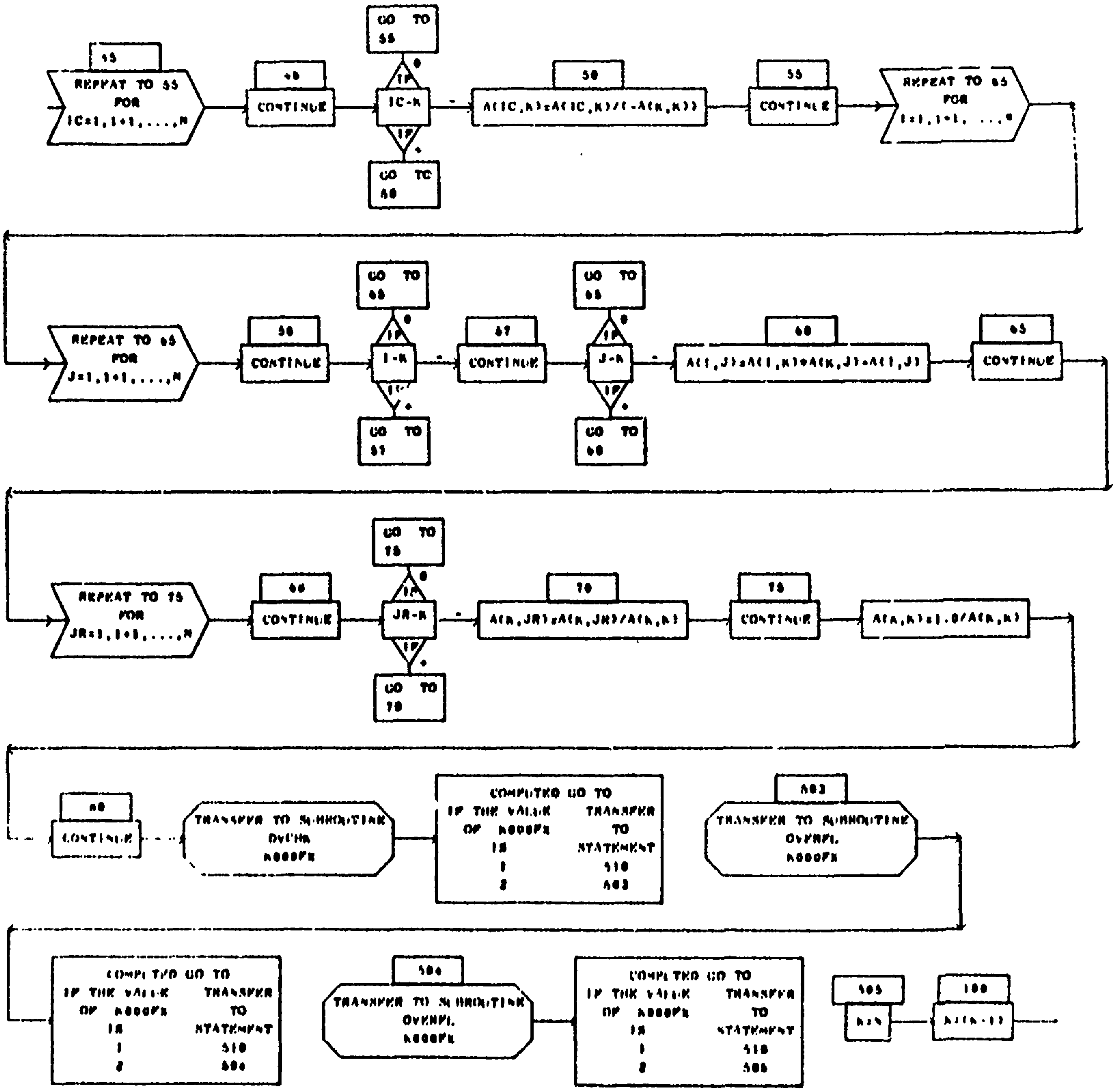
RETURN

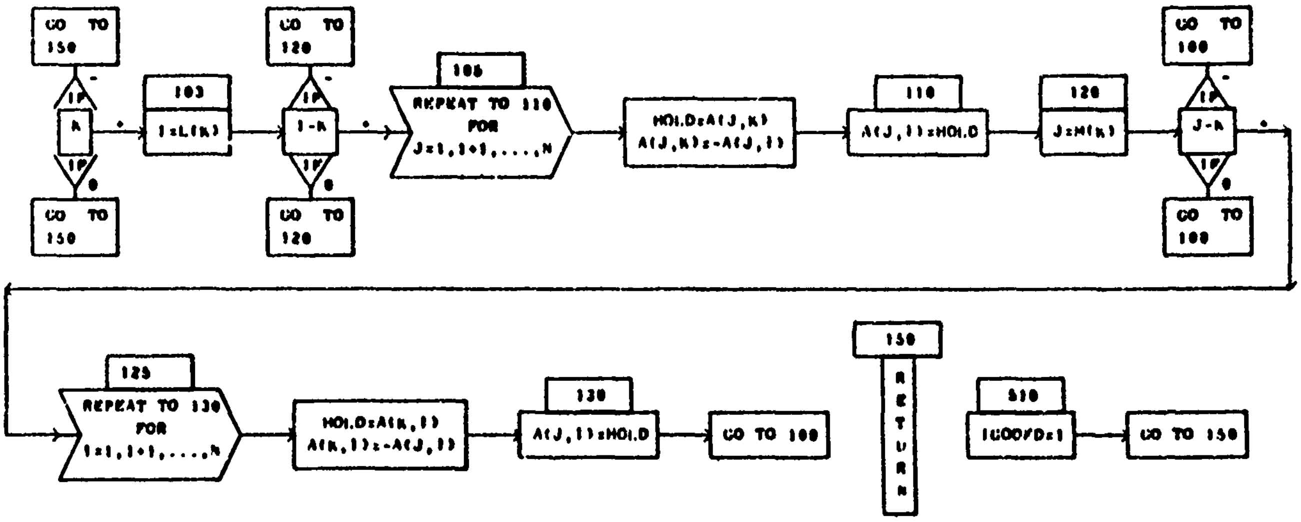
INVERS

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	6,6	L	6	M	6				


START
TRANSFER TO SUBROUTINE OVERFL KOOOFX
COMPUTED GO TO
IF THE VALUE OF KOOOFX IS
TRANSFER TO STATEMENT
1 500
2 500
500
TRANSFER TO SUBROUTINE OVERFL KOOOFX
COMPUTED GO TO
IF THE VALUE OF KOOOFX IS
TRANSFER TO STATEMENT
1 501
2 501
501
TRANSFER TO SUBROUTINE DVCHK KOOOFX
COMPUTED GO TO
IF THE VALUE OF KOOOFX IS
TRANSFER TO STATEMENT
1 502
2 502
502
IGOOFD: 0
REPEAT TO 40 FOR K=1,1+1,...,N
L(K)=K
M(K)=K
BIGA=A(K,K)
REPEAT TO 20 FOR I=K,K+1,...,N
REPEAT TO 20 FOR J=K,K+1,...,N
ABS(BIGA)-ABS(A(I,J))
GO TO 20
GO TO 20
10
BIGA=A(I,J)
L(K)=I
M(K)=J
20
CONTINUE
JROW=L(K)
L(K)-K
GO TO 35
GO TO 35
25
REPEAT TO 30 FOR I=1,1+1,...,N
HOLD=-A(K,I)
A(K,I)=A(JROW,I)
30
A(JROW,I)=HOLD
35
ICOL=M(K)
M(K)-K
GO TO 45
GO TO 45
37
REPEAT TO 40 FOR J=1,1+1,...,N
HOLD=-A(J,K)
A(J,K)=A(J,ICOL)
40
A(J,ICOL)=HOLD

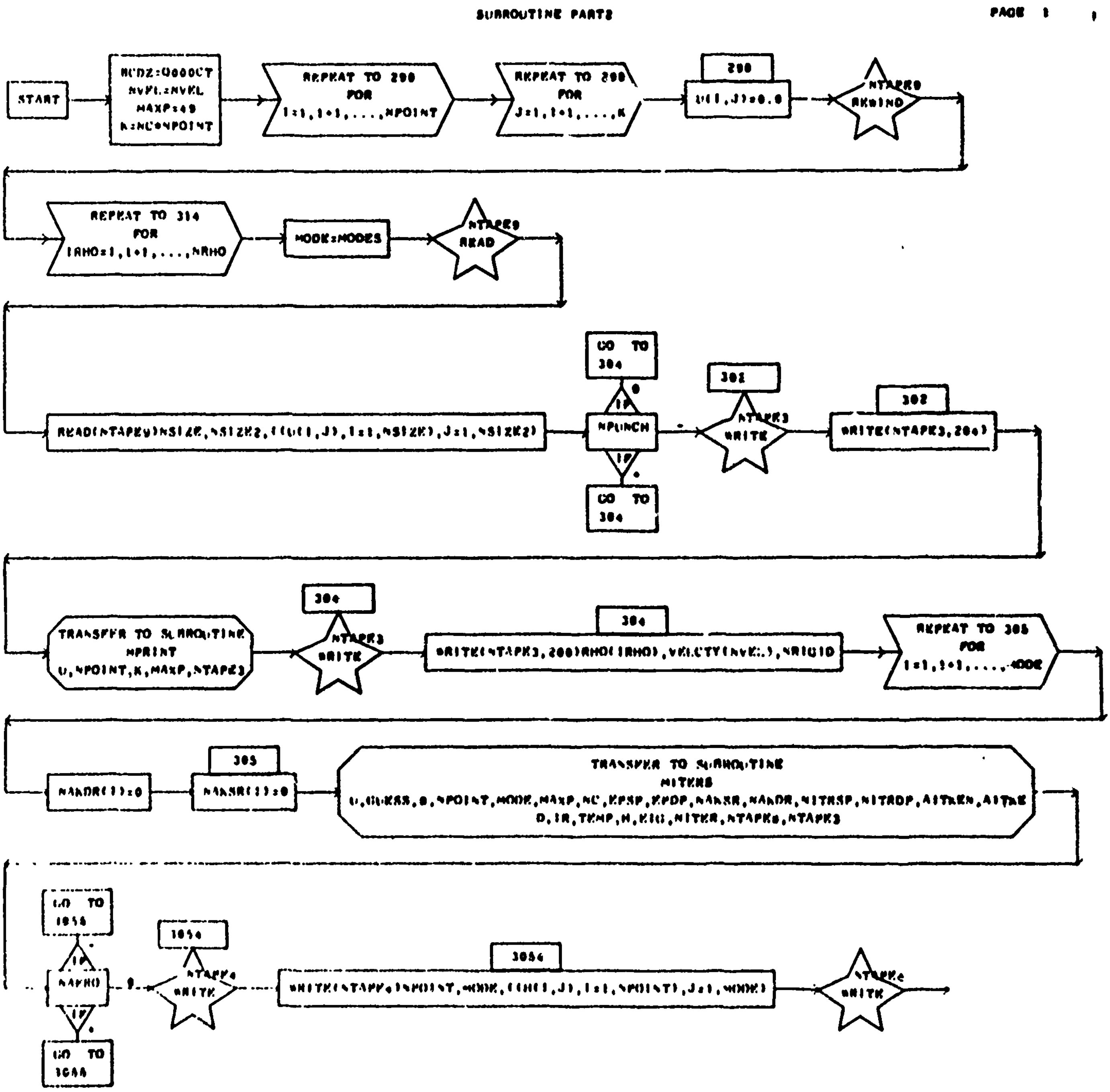

45
REPEAT TO 55 FOR
CONTINUE
GO TO 55
IF
IC-K
GO TO
50
A(IC,K)=A(IC,K)/(-A(K,K))
55
CONTINUE
REPEAT TO 65 FOR
REPEAT TO 65 FOR
CONTINUE
GO TO 65
I-K
CONTINUE
GO TO 65
J-K
A(I,J)=A(I,K)*A(K,J)+A(I,J)
CONTINUE
GO TO
GO TO
REPEAT TO 75 FOR
CONTINUE
GO TO 75
JR-K
70
A(K,JR)=A(K,JR)/A(K,K)
75
CONTINUE
A(K,K)=1.0/A(K,K)
GO TO 70
CONTINUE
TRANSFER TO SUBROUTINE
COMPUTED GO TO
IF THE VALUE TRANSFER
OF TO
IS STATEMENT
1
2
TRANSFER TO SUBROUTINE
COMPUTED GO TO
IF THE VALUE TRANSFER
OF TO
IS STATEMENT
1
2
TRANSFER TO SUBROUTINE
COMPUTED GO TO
IF THE VALUE TRANSFER
OF TO
IS STATEMENT
1
2

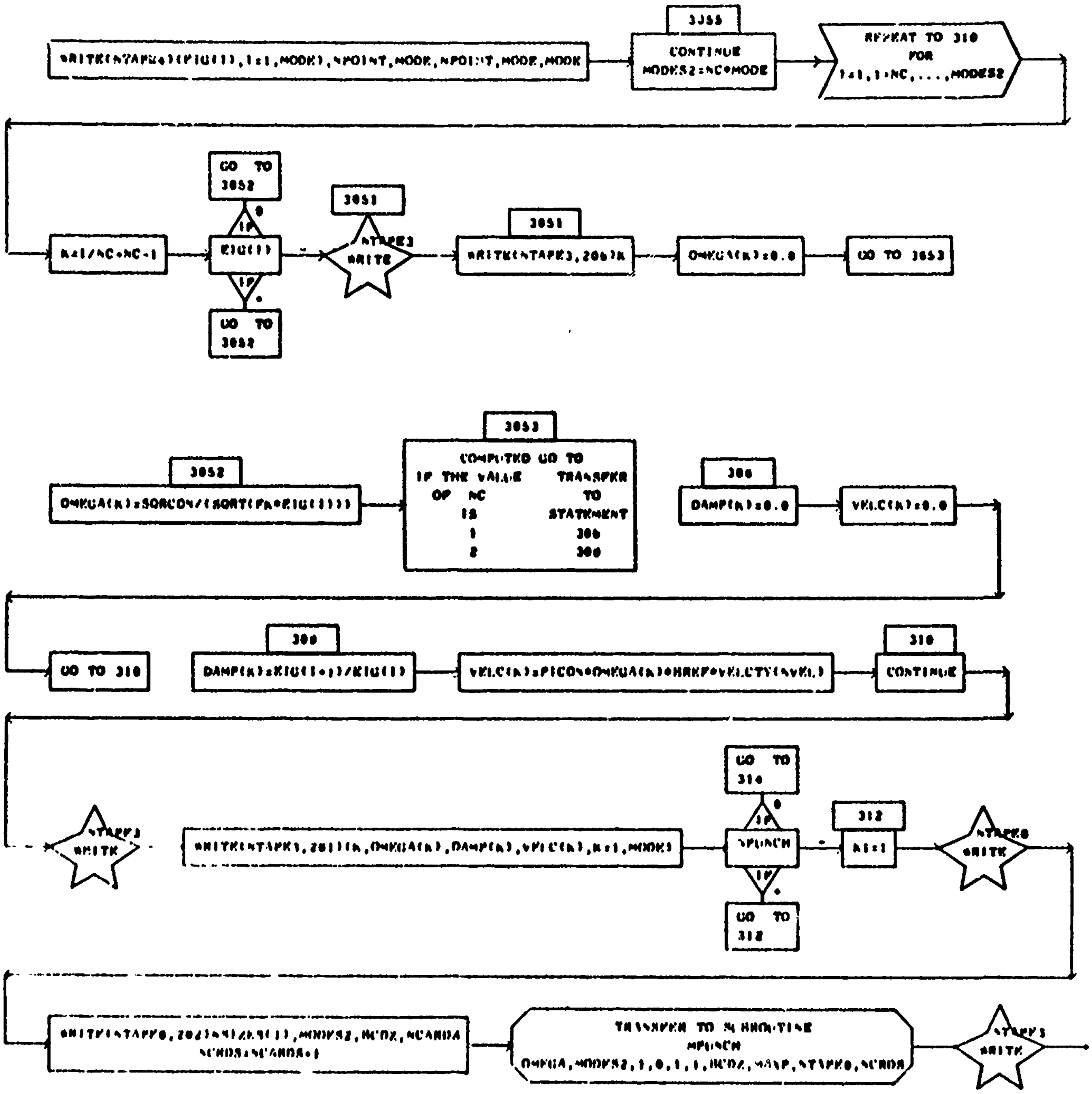

GO TO 150
IF
K
IF
GO TO 150
103
I=L(K)
GO TO 120
IF
I-K
IF
GO TO 120
105
REPEAT TO 110
FOR
J=1,1+1,....,N
HOLD=A(J,K)
A(J,K)=-A(J,I)
110
A(J,I)=HOLD
120
J=M(K)
GO TO 100
IF
J-K
IF
GO TO 100
125
REPEAT TO 130
FOR
I=1,I+1,....,N
HOLD=A(K,I)
A(K,I)=-A(J,I)
130
A(J,I)=HOLD
GO TO 100
150
RETURN
510
IGOOFD=1
GO TO 150

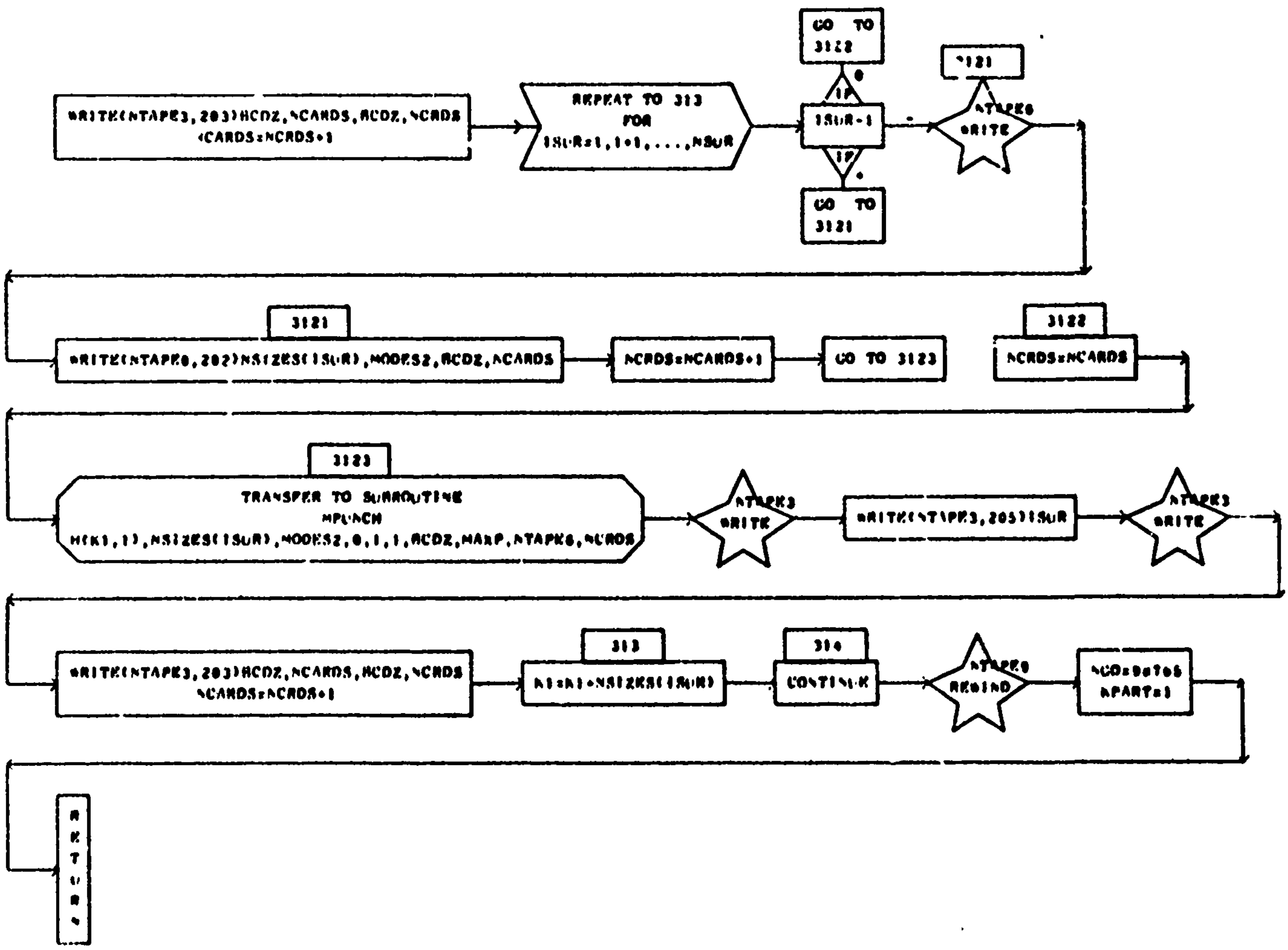
PART2

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
IT	216	VELCTY	20	NSIZES	20	DM	6,6	RHO	20


SUBROUTINE PART2
PAGE 1
START
MAXP=49
REPEAT TO 299 FOR I=1,1+1,...,NPOINT
REPEAT TO 299 FOR J=1,1+1,...,K
299
NTAPE9 REWIND
REPEAT TO 314 FOR IRHO=1,1+1,...,NRHO
MODE=MODES
NTAPE9 READ
GO TO 304
IF
NPUNCH
301
NTAPE3 WRITE
302
TRANSFER TO SUBROUTINE NPRINT
U,NPOINT,K,MAXP,NTAPE3
304
NTAPE3 WRITE
REPEAT TO 305 FOR
305
TRANSFER TO SUBROUTINE
GO TO
WRITE

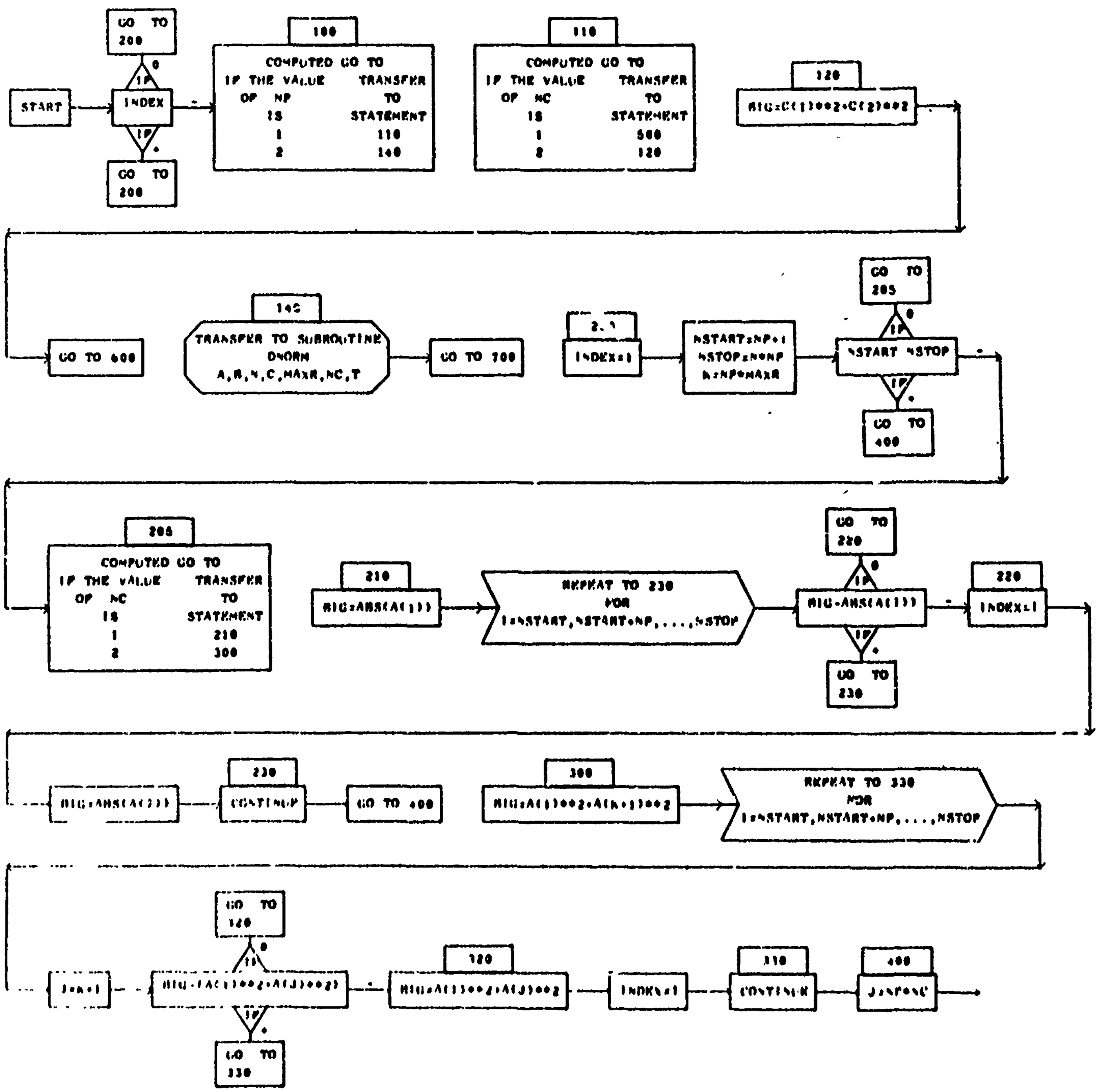

WRITE(NTAPE4)(EIG(I),I=1,MODE),NPOINT,MODE,NPOINT,MODE,MODE
3355
CONTINUE
MODES2=NC*MODE
REPEAT TO 310
FOR
K=1,1+NC,...,MODES2
K=I/NC+NC-1
GO TO
3052
IF
EIG(I)
IF
GO TO
3052
3051
NTAPE3
WRITE
3051
WRITE(NTAPE3,206)K
OMEGA(K)=0.0
GO TO 3053
3052
OMEGA(K)=SQRCON/(SQRT(FK*EIG(I)))
3053
COMPUTED GO TO
IF THE VALUE TRANSFER
OF NC TO
IS STATEMENT
1 306
2 308
306
DAMP(K)=0.0
VELC(K)=0.0
GO TO 310
308
DAMP(K)=EIG(I+1)/EIG(I)
VELC(K)=PICON*OMEGA(K)*BREF*VELCTY(NVEL)
310
CONTINUE
NTAPE3
WRITE
WRITE(NTAPE3,201)(K,OMEGA(K),DAMP(K),VELC(K),K=1,MODE)
GO TO
316
IF
NPUNCH
IF
GO TO
312
312
KI=1
NTAPE6
WRITE
WRITE(NTAPE6,202)NSIZE,S(1),MODES2,NCDZ,NCARDA
NCRDS=NCARDS+1
TRANSFER TO SUBROUTINE
MPUNCH
OMEGA,MODES2,1,0,1,1,NCDZ,NAMP,NTAPE6,NCRDS
NTAPE3
WRITE

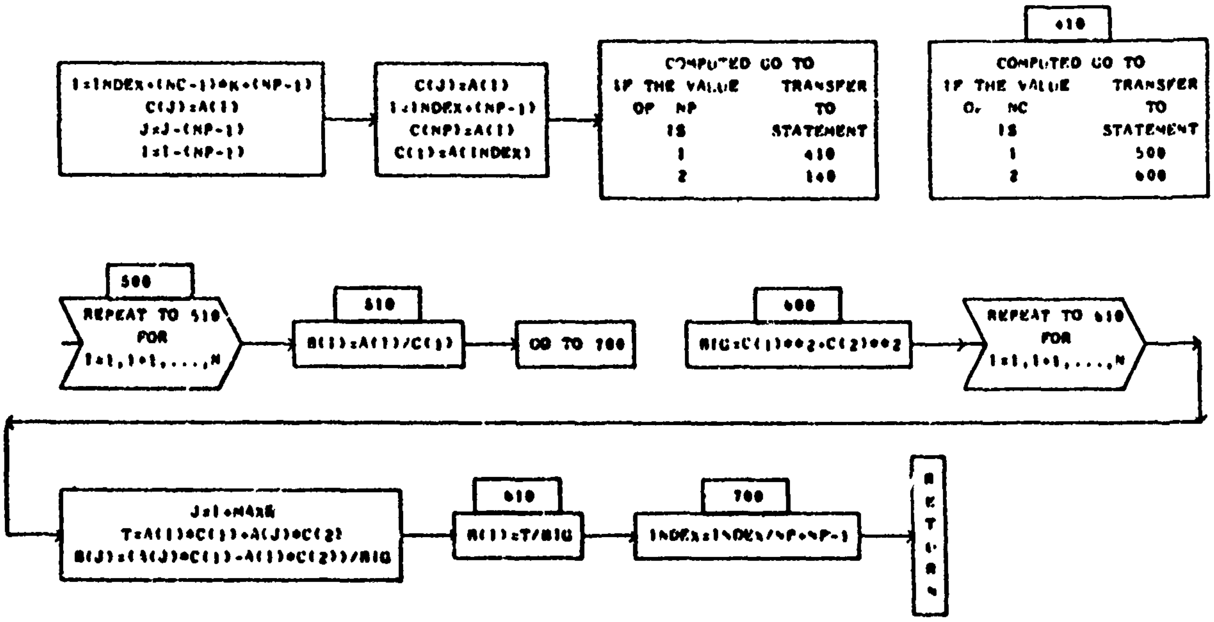

WRITE(NTAPE3,203)RCDZ,NCARDS,RCDZ,NCRDS
NCARDS=NCRDS+1
REPEAT TO 313
FOR
ISUR=1,1+1,...,NSUR
GO TO
3122
IF
ISUR-1
IF
GO TO
3121
3121
NTAPE6
WRITE
3121
WRITE(NTAPE6,202)NSIZES(ISUR),NODES2,RCDZ,NCARDS
NCRDS=NCARDS+1
GO TO 3123
3122
NCRDS=NCARDS
3123
TRANSFER TO SUBROUTINE
NPUNCH
H(K1,1),NSIZES(ISUR),NODES2,0,1,1,RCDZ,MAXP,NTAPE6,NCRDS
NTAPE3
WRITE
WRITE(NTAPE3,205)ISUR
NTAPE3
WRITE
WRITE(NTAPE3,203)RCDZ,NCARDS,RCDZ,NCRDS
NCARDS=NCRDS+1
313
K1=K1+NSIZES(ISUR)
314
CONTINUE
NTAPE6
REWIND
NCD=0
NPART=1
RETURN

NORM2

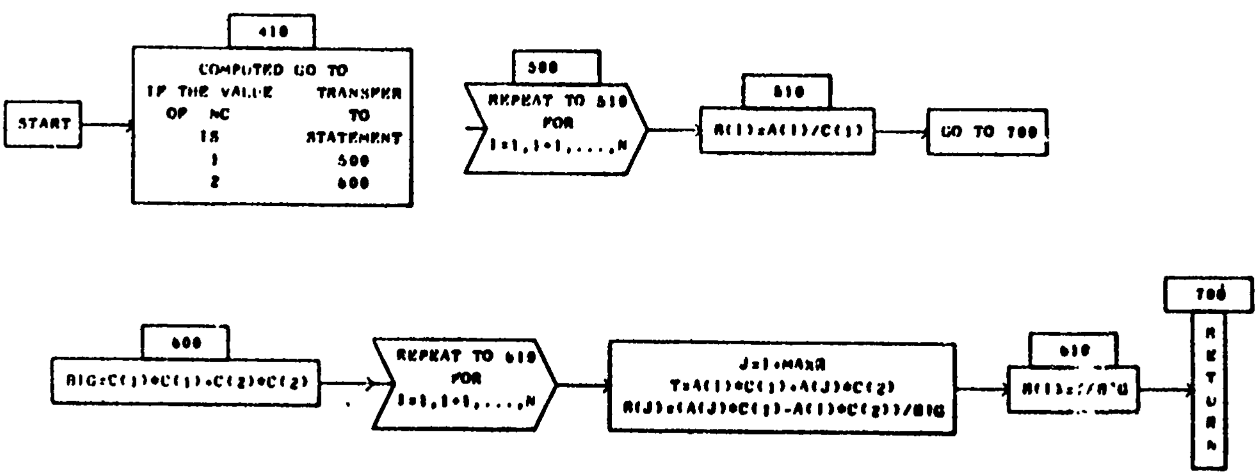
DIMENSIONED VARIABLES

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	B	1	C	1	T	2		


START
INDEX
GO TO 200
GO TO 200
100
COMPUTED GO TO
IF THE VALUE OF NP IS 1 2 TRANSFER TO STATEMENT 110 140
110
COMPUTED GO TO
IF THE VALUE OF NC IS 1 2 TRANSFER TO STATEMENT 500 120
120
GO TO 600
140
TRANSFER TO SUBROUTINE DNORM A,B,N,C,MAXR,NC,T
GO TO 700
INDEX=1
NSTART=NP+1
NSTOP=N*NP
K=NP*MAXR
NSTART NSTOP
GO TO 205
GO TO 400
205
COMPUTED GO TO
IF THE VALUE OF NC IS 1 2 TRANSFER TO STATEMENT 210 300
210
REPEAT TO 230 FOR I=NSTART,NSTART+NP,...,NSTOP
GO TO 220
GO TO 230
220
INDEX=I
230
CONTINUE
GO TO 400
300
REPEAT TO 330 FOR I=NSTART,NSTART+NP,...,NSTOP
GO TO 320
GO TO 330
320
INDEX=I
330
CONTINUE
400


I=INDEX+(NC-1)*K+(NP-1)
C(J)=A(I)
J=J-(NP-1)
I=I-(NP-1)
C(J)=A(I)
I=INDEX+(NP-1)
C(NP)=A(I)
C(1)=A(INDEX)
COMPUTED GO TO
IF THE VALUE OF NP IS 1 TRANSFER TO STATEMENT 410
2 140
410
COMPUTED GO TO
IF THE VALUE OF NC IS 1 TRANSFER TO STATEMENT 500
2 600
500
REPEAT TO 510 FOR I=1,I+1,...,N
510
B(I)=A(I)/C(1)
GO TO 700
600
BIG=C(1)**2+C(2)**2
REPEAT TO 610 FOR I=1,I+1,...,N
J=I+MAXB
T=A(I)*C(1)+A(J)*C(2)
B(J)=(A(J)*C(1)-A(I)*C(2))/BIG
610
B(I)=T/BIG
700
INDEX=INDEX/NP+NP-1
RETURN

DNORMZ

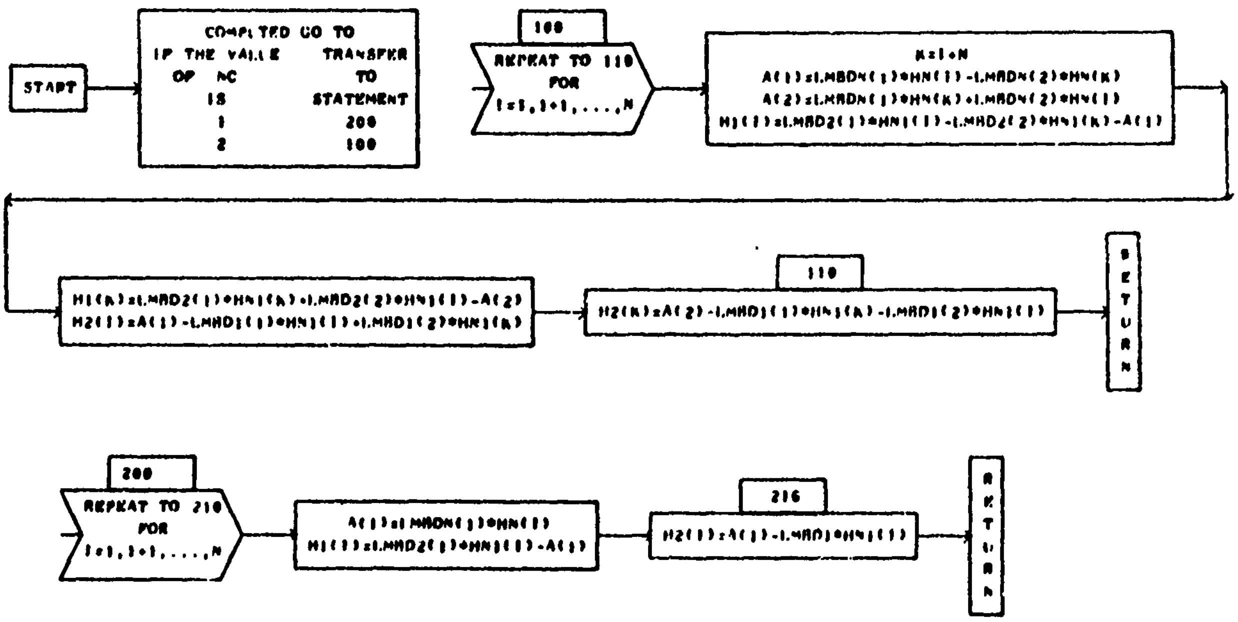

START
410
COMPUTED GO TO
IF THE VALUE OF NC IS
TRANSFER TO STATEMENT
1 500
2 600
500
REPEAT TO 510 FOR
I=1,1+1,....,N
510
B(I)=A(I)/C(1)
GO TO 700
600
BIG=C(1)*C(1)+C(2)*C(2)
REPEAT TO 610 FOR
I=1,1+1,....,N
J=I+MAXR
T=A(I)*C(1)+A(J)*C(2)
B(J)=(A(J)*C(1)-A(I)*C(2))/BIG
610
B(I)=T/BIG
700
RETURN

POHS

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
LMBDN	1	LMBD1	2	LMBD2	1	HN	1	HN1	1
H1	1	H2	1	A	2				

SUBROUTINE POH (LMBDN,LMBD1,LMBD2,HN,HN1,H1,H2,N,NC) PAGE 1

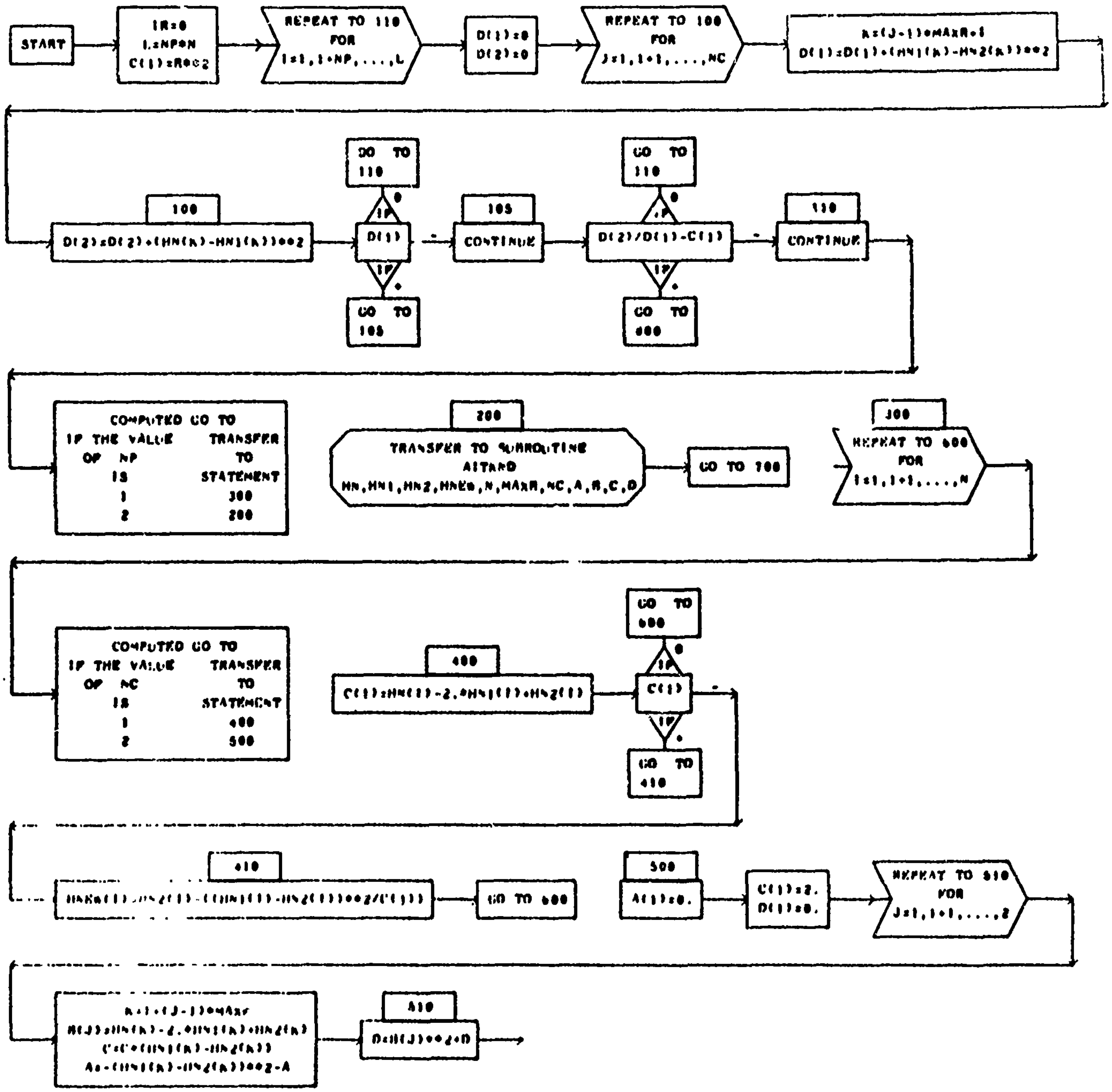


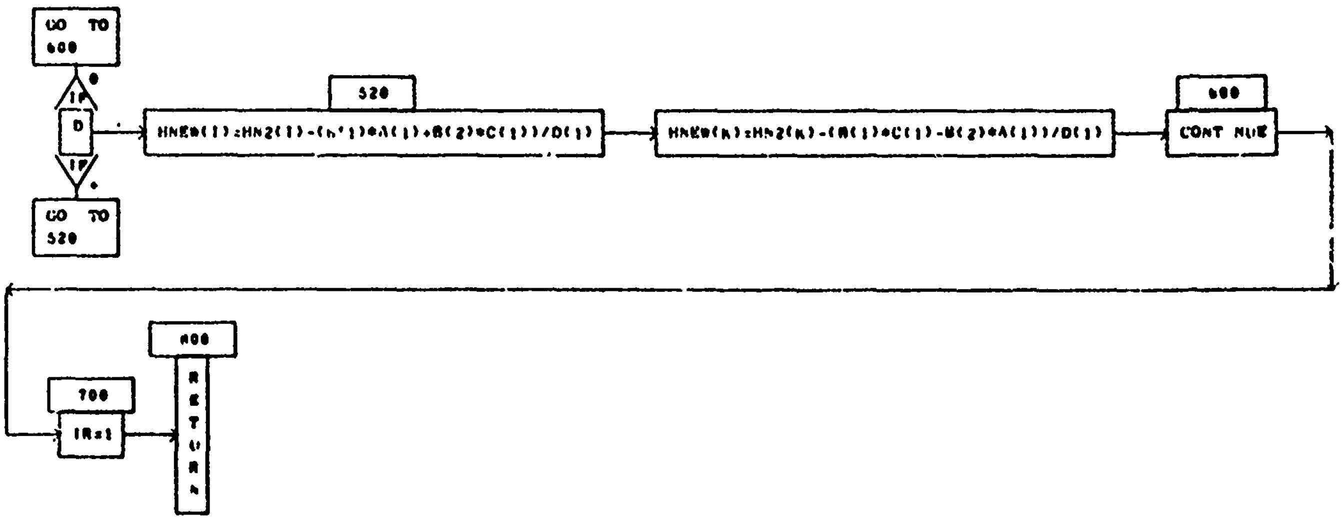
AITKNS

DIMENSIONED VARIABLES

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
HN	1	HN1	1	HN2	1	HNEW	1	A	2
B	4	C	2	D	2				

SUBROUTINE AITKNS (HN,HN1,HN2,HNEW,R,N,MAXR,NC,NP,IR) PAGE 1

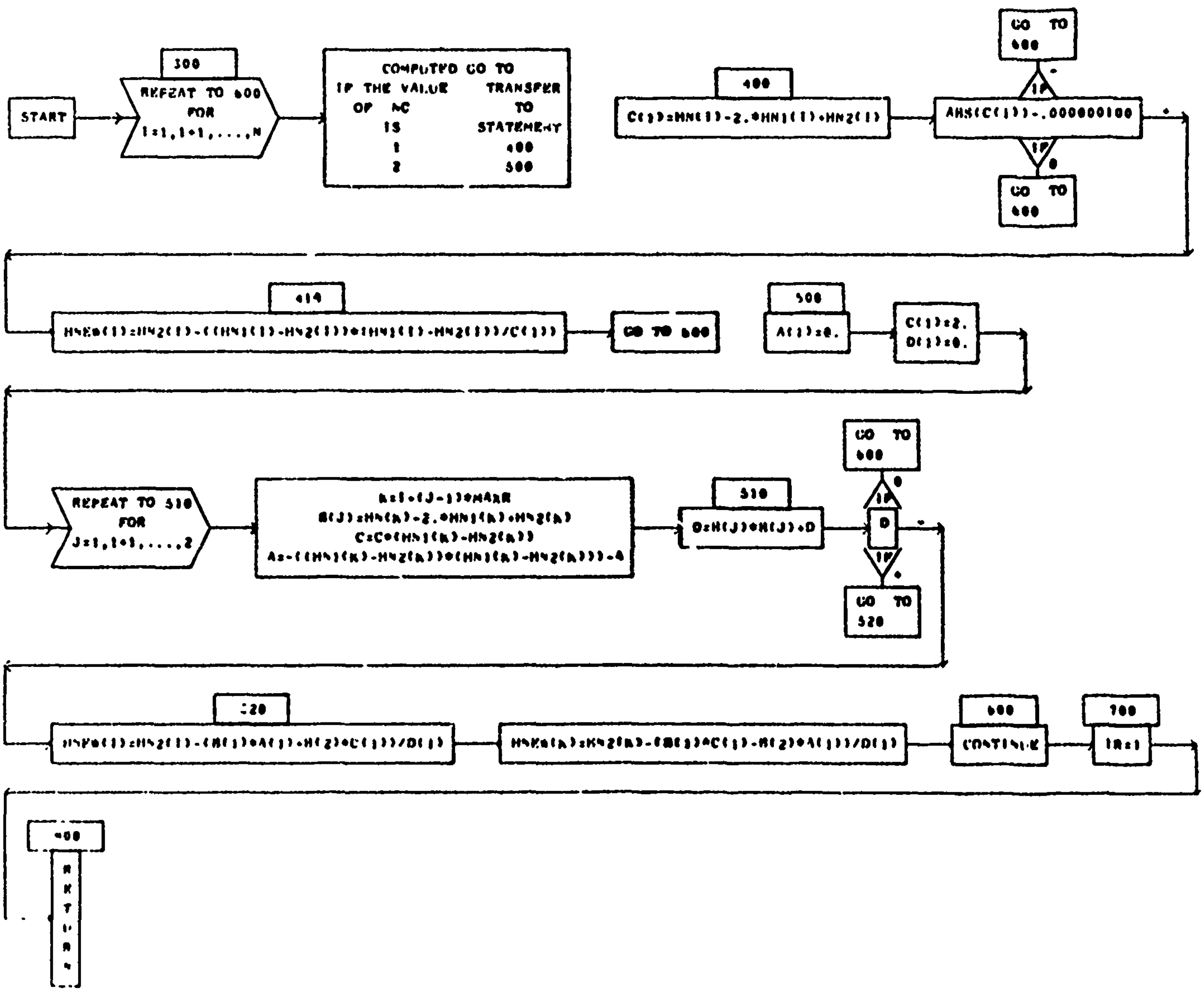



GO TO 600
IF
0
D
IF
+
GO TO 520
520
HNEW(I)=HN2(I)-(B(1)*A(1)+B(2)*C(1))/D(1)
HNEW(K)=HN2(K)-(B(1)*C(1)-B(2)*A(1))/D(1)
600
CONTINUE
700
IR=1
800
RETURN

AITKND

D I M E N S I O N E D   V A R I A B L E S

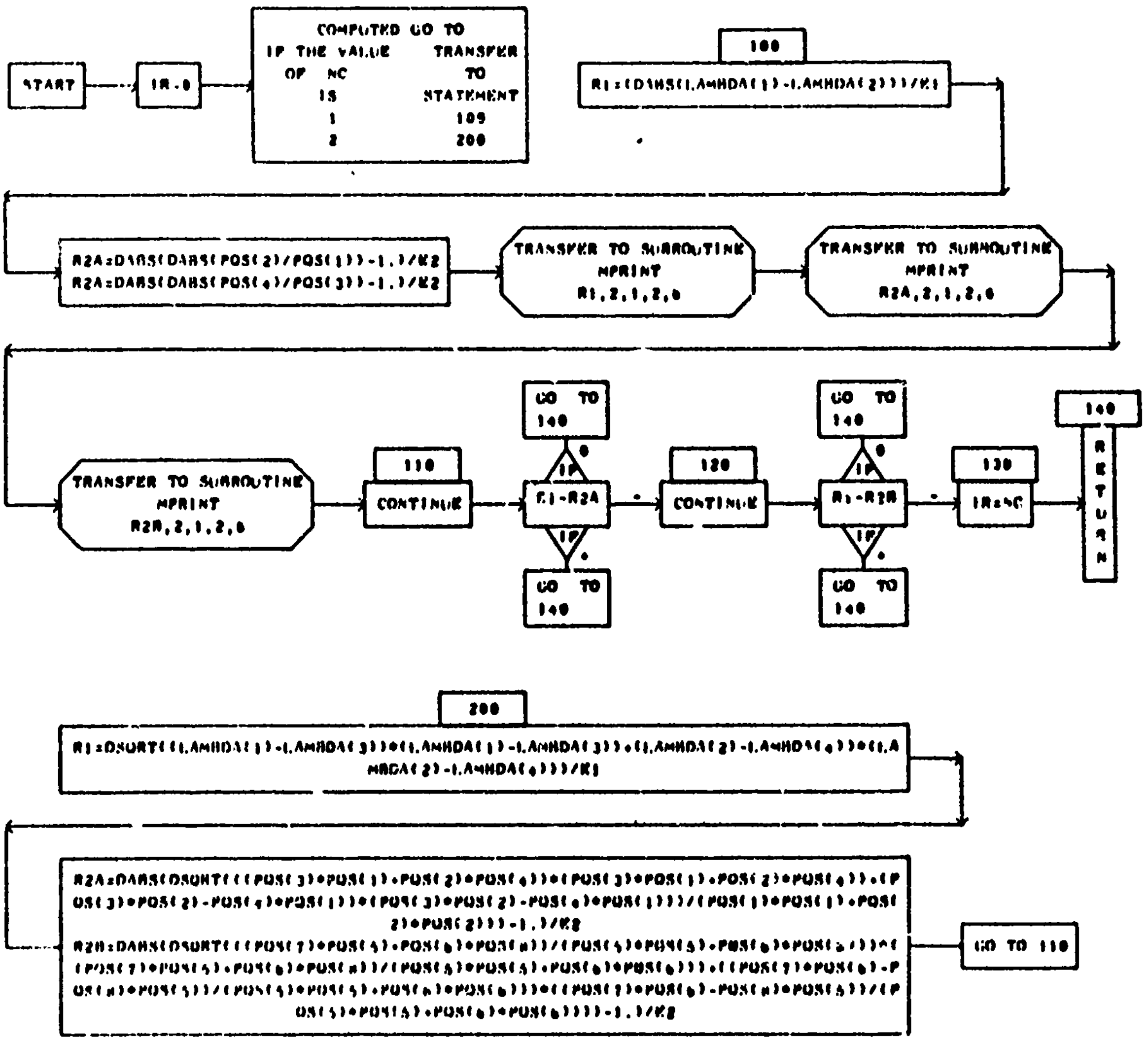
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
HN	1	HN1	1	HN2	1	HNEW	1	A	1
B	2	C	1	D	1				


START
300
REPEAT TO 600 FOR I=1,I+1,...,N
COMPUTED GO TO
IF THE VALUE OF NC IS 1 2
TRANSFER TO STATEMENT 400 500
400
C(I)=HN(I)-2.*HN1(I)+HN2(I)
ABS(C(I))-.000000100
GO TO 600
GO TO 600
410
HNEW(I)=HN2(I)-((HN1(I)-HN2(I))*(HN1(I)-HN2(I))/C(I))
GO TO 600
500
A(I)=0.
C(I)=2.
D(I)=0.
REPEAT TO 510 FOR J=1,1+1,...,2
510
D=B(J)*B(J)+D
GO TO 600
GO TO 520
520
600
CONTINUE
700
IR=1
RETURN

LEGISS

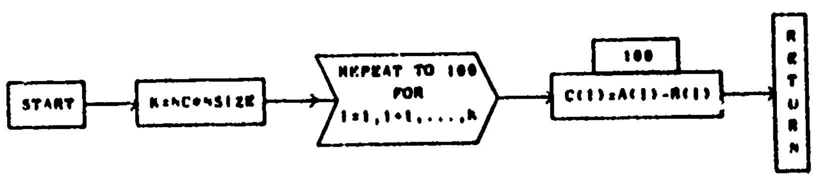
D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
LAMBDA	1	POS	1						


START
IR=0
COMPUTED GO TO
IF THE VALUE OF NC IS 1 TRANSFER TO STATEMENT 100
2 200
100
R1=(DABS(LAMBDA(1)-LAMBDA(2)))/E1
R2A=DABS(DABS(POS(2)/POS(1))-1.)/E2
R2A=DABS(DABS(POS(4)/POS(3))-1.)/E2
TRANSFER TO SUBROUTINE MPRINT R1,2,1,2,6
TRANSFER TO SUBROUTINE MPRINT R2A,2,1,2,6
TRANSFER TO SUBROUTINE MPRINT R2B,2,1,2,6
110
CONTINUE
E1-R2A
120
CONTINUE
R1-R2B
130
IR=NC
GO TO 140
140
RETURN
200
GO TO 110

MADS

SUBROUTINE MADS (A,B,C,NSIZE,NC) PAGE 1

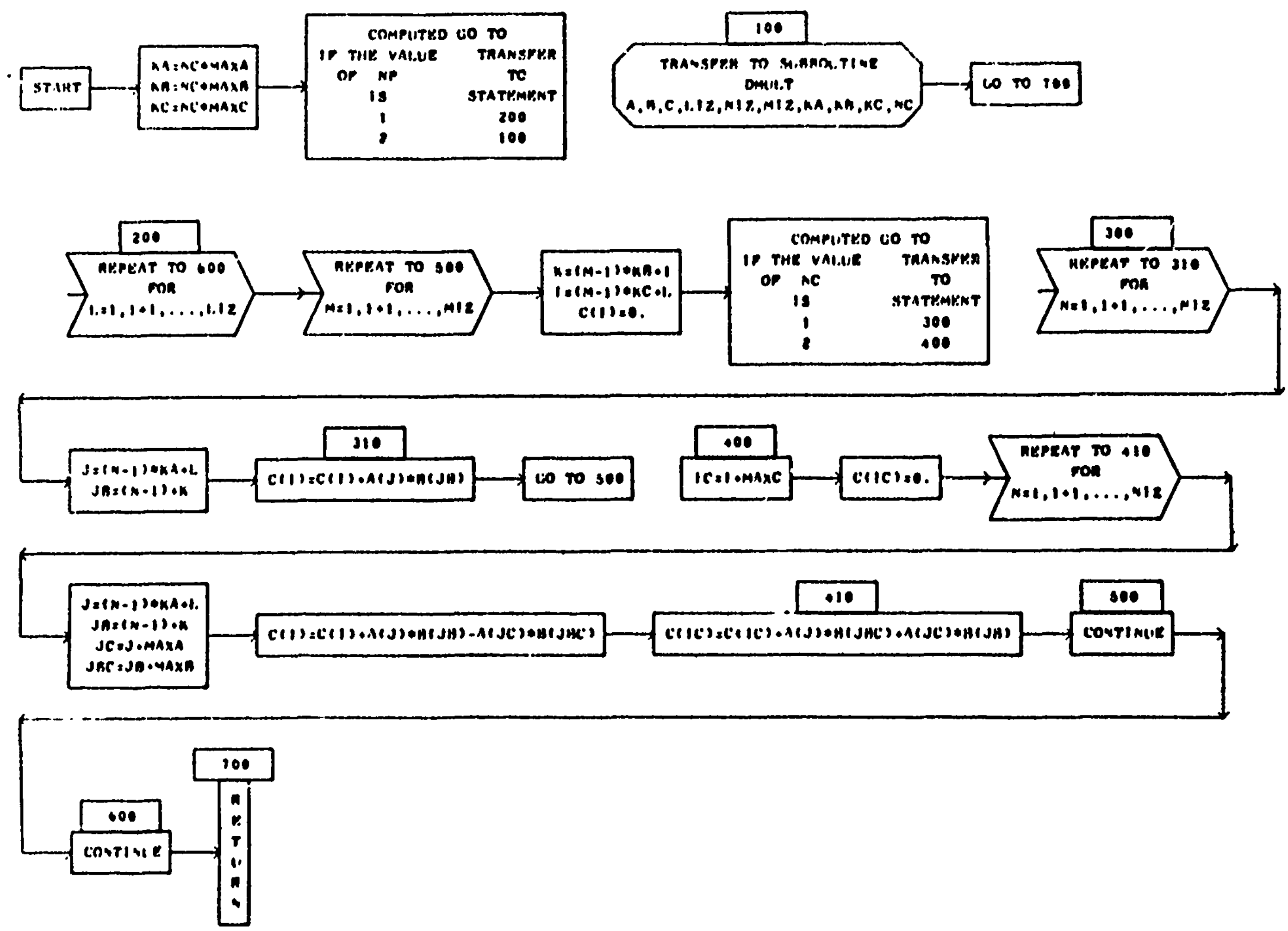


MULTS

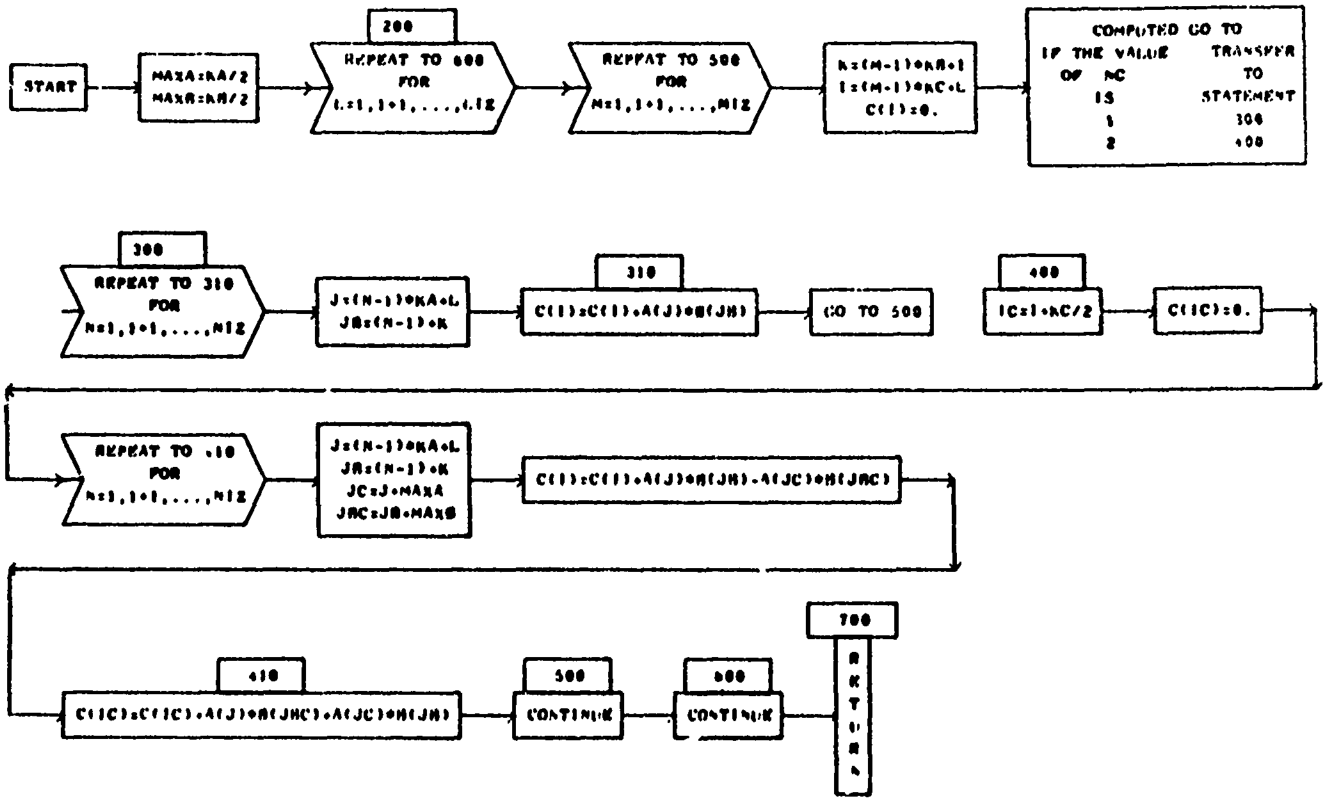
D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	B	1	C	1				

SUBROUTINE MULT (A,B,C,L1Z,N1Z,M1Z,MAXA,MAXB,MAXC,NC,NP) PAGE 1



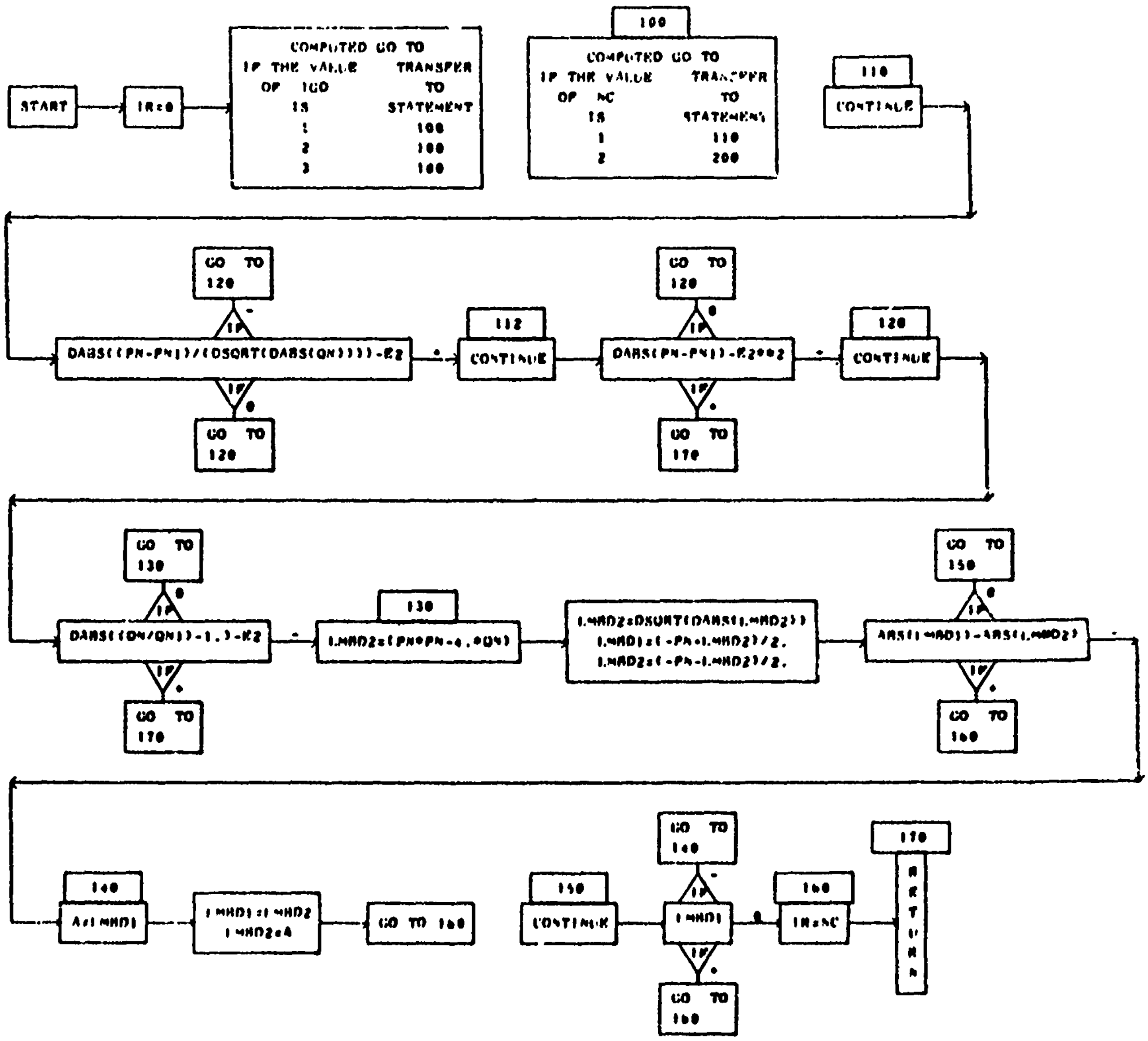
SUBROUTINE DMULT (A,B,C,LIZ,MIZ,NIZ,KA,KB,KC,NC) PAGE 1

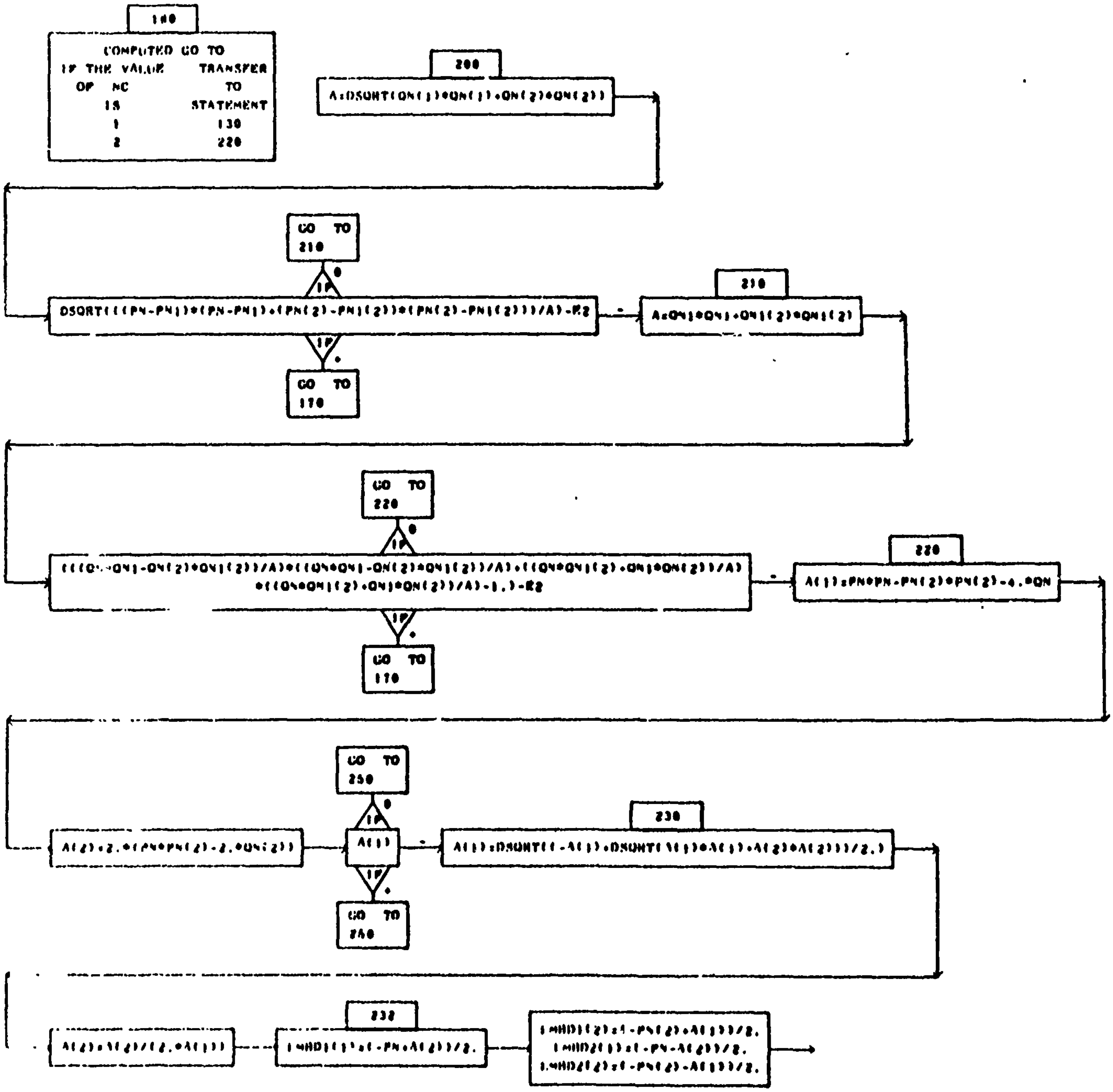


POLMS

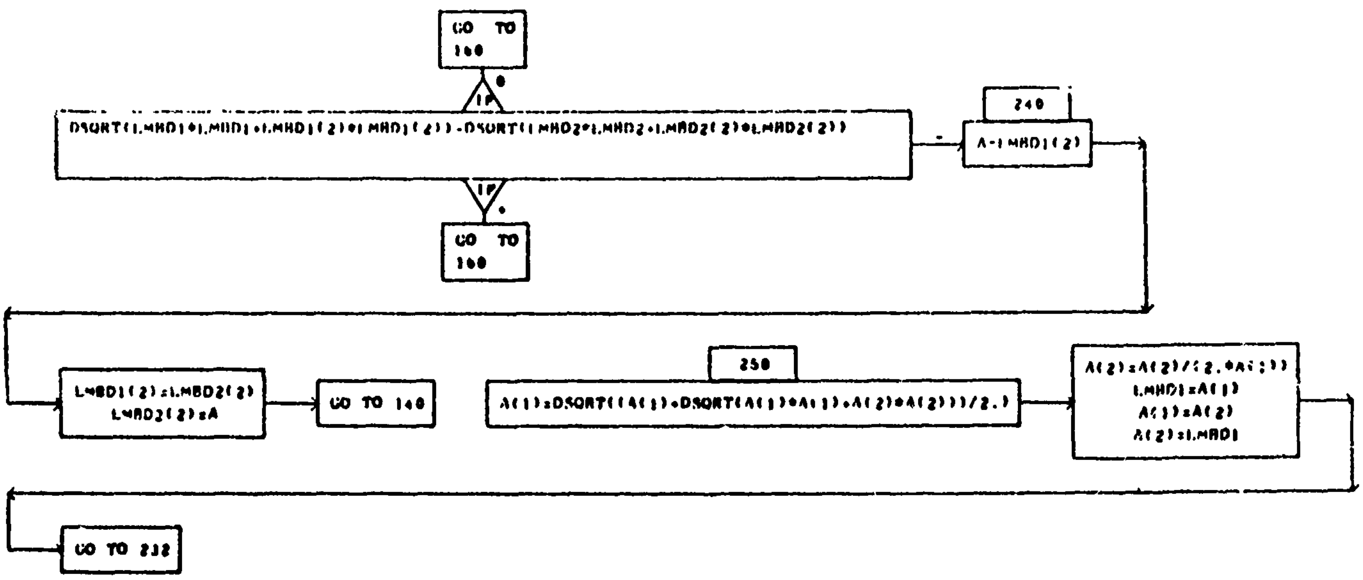
D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
PN	1	PN1	1	QN	1	QN1	1	E2	1
LMBD1	1	LMBD2	1						


START
IR=0
COMPUTED GO TO
IF THE VALUE OF IGO IS
TRANSFER TO STATEMENT
100
CONTINUE
110
120
130
140
150
160
170
GO TO 120
GO TO 130
GO TO 150
GO TO 170
GO TO 140
GO TO 160
IF
DABS((PN-PN1)/(DSQRT(DABS(QN))))-E2
DABS(PN-PN1)-E2**2
DABS((QN/QN1)-1.)-E2
LMRD2=(PN*PN-4.*QN)
LMRD2=DSQRT(DABS(LMRD2))
LMRD1=(-PN+LMRD2)/2.
LMRD2=(-PN-LMRD2)/2.
ABS(LMRD1)-ABS(LMRD2)
A=LMRD1
LMRD1=LMRD2
LMRD2=A
GO TO 160
IR=NC
RETURN



140
COMPUTED GO TO
IF THE VALUE OF NC IS 1 TRANSFER TO STATEMENT 130
2 220
200
GO TO 210
IF
GO TO 170
210
GO TO 220
220
GO TO 170
GO TO 250
A(1)
230
GO TO 260
232

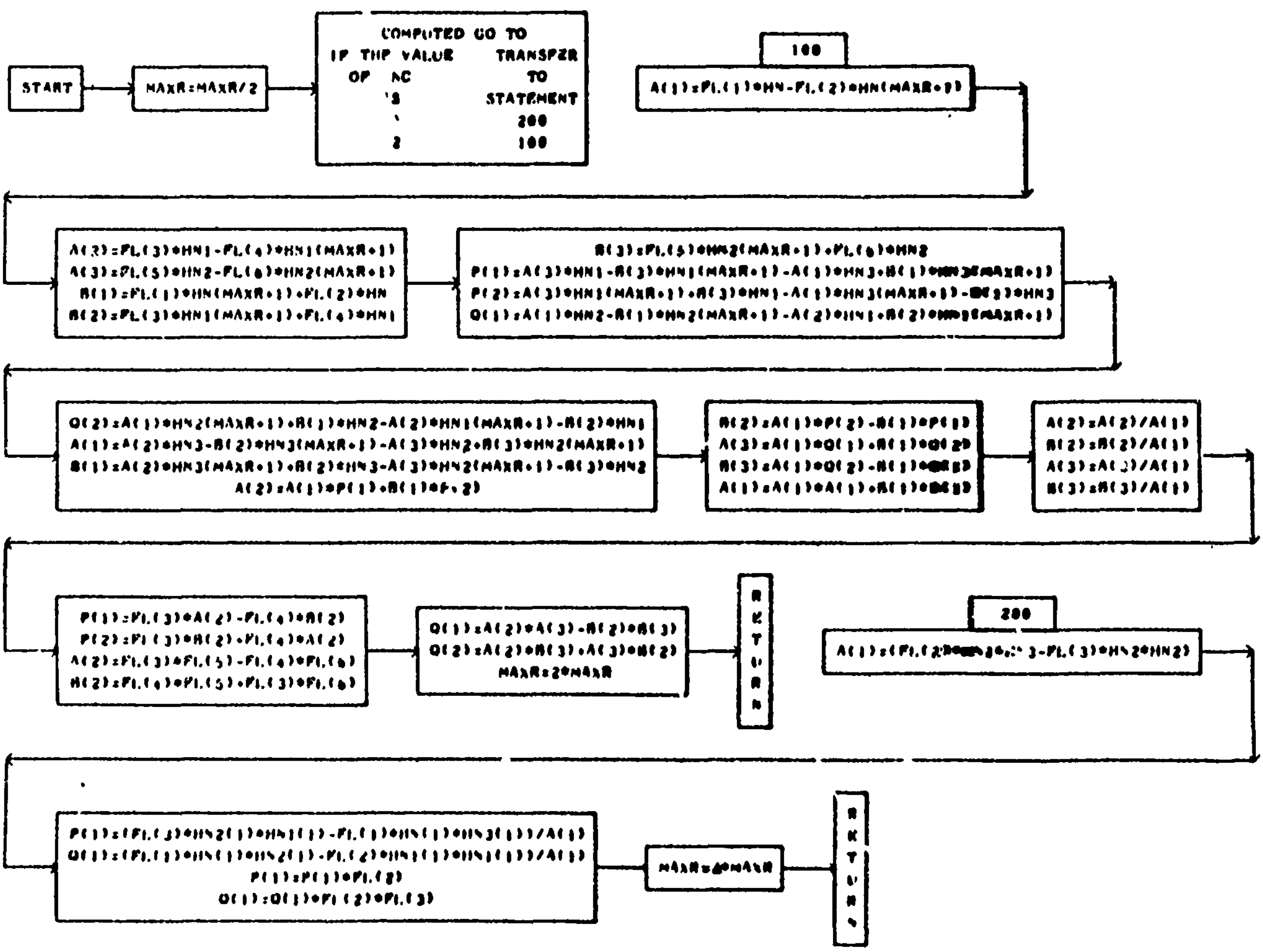

GO TO 160
0
IF
DSQRT(LMBD1*LMBD1+LMBD1(2)*LMBD1(2))-DSQRT(LMBD2*LMBD2+LMBD2(2)*LMBD2(2))
IF
+
GO TO 160
-
240
A=LMBD1(2)
LMBD1(2)=LMBD2(2)
LMBD2(2)=A
GO TO 140
250
A(1)=DSQRT((A(1)+DSQRT(A(1)*A(1)+A(2)*A(2)))/2.)
A(2)=A(2)/(2.*A(1))
LMBD1=A(1)
A(1)=A(2)
A(2)=LMBD1
GO TO 232

PUS

D I M E N S I O N E D   V A R I A B L E S

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
FL	1	HN	1	HN1	1	HN2	1	HN3	1
P	1	Q	1	A	3	B	3		

SUBROUTINE PQ (FL, HN, HN1, HN2, HN3, P, Q, NC, MAXR) PAGE 1



CLOSES

SUBROUTINE CLOSE, COMPUTES 2 CLOSE ROOTS.

U = MATRIX, DIMENSIONED (MAXR,2*NC*MAXR)

H = STARTING GUESS, DIMENSIONED((MAXR,2*NC*4)+2*NC*N)

NSIZE = SIZE OF MATRIX

MAXR = DIMENSIONED NUMBER OF ROWS OF U AND H

MAXTRY = MAXIMUM NUMBER OF DOUBLE PRECISION ITERATIONS.

EPS1 = SINGLE ROOT CONVERGENCE CRITERIA

EPS2 = DOUBLE ROOT CONVERGENCE CRITERIA

R= AITKENS CONVERGENCE CRITERIA

IRR = ERROR RETURN INDICATER. =1, OVERFLOW

=2, DIVIDE CHECK

=3, BOTH OVERFLOW AND DIVIDE CHECK.

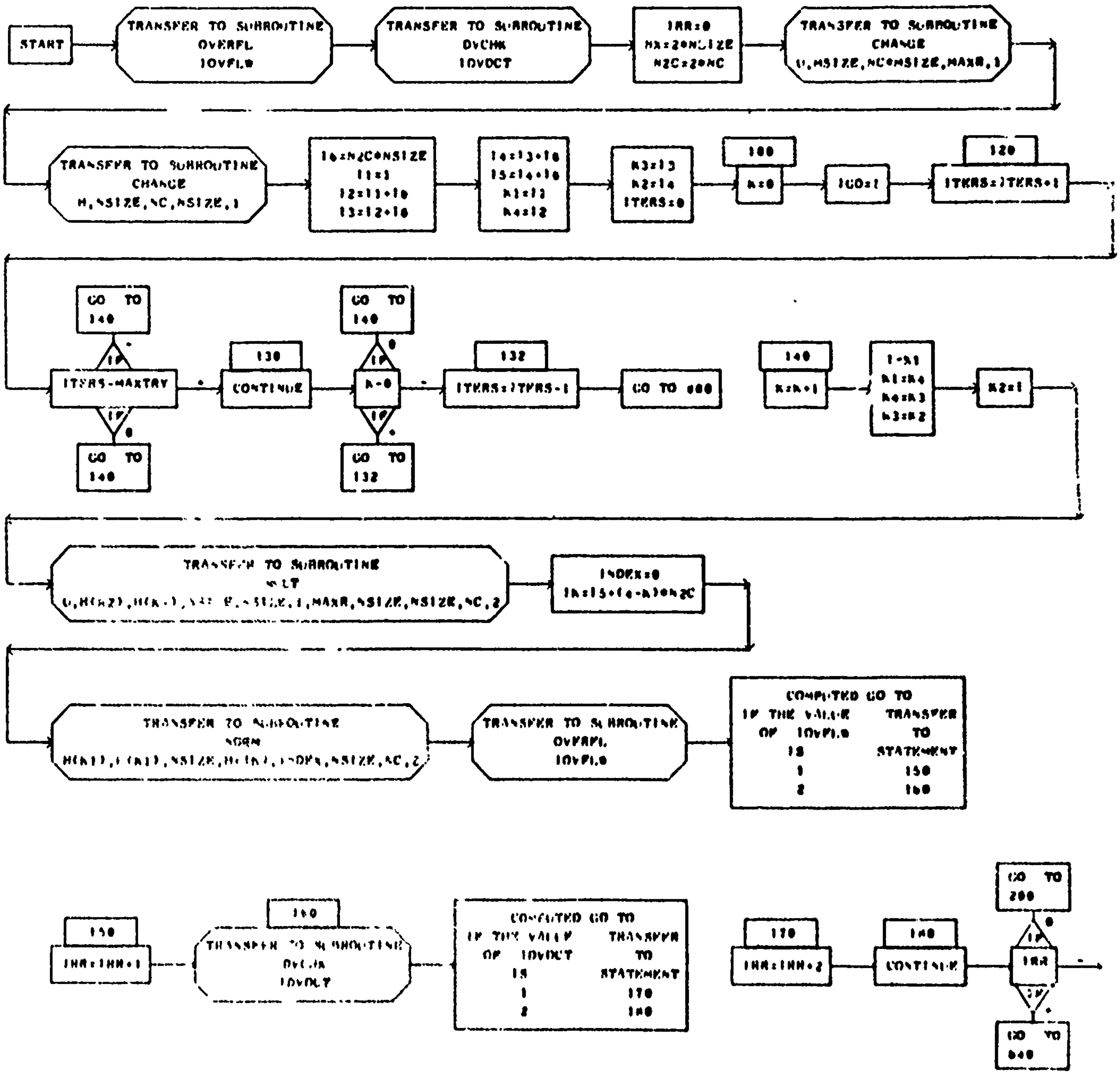
ITERS= NUMBER OF ITERATIONS PERFORMED, - FOR DOUBLE ROOT

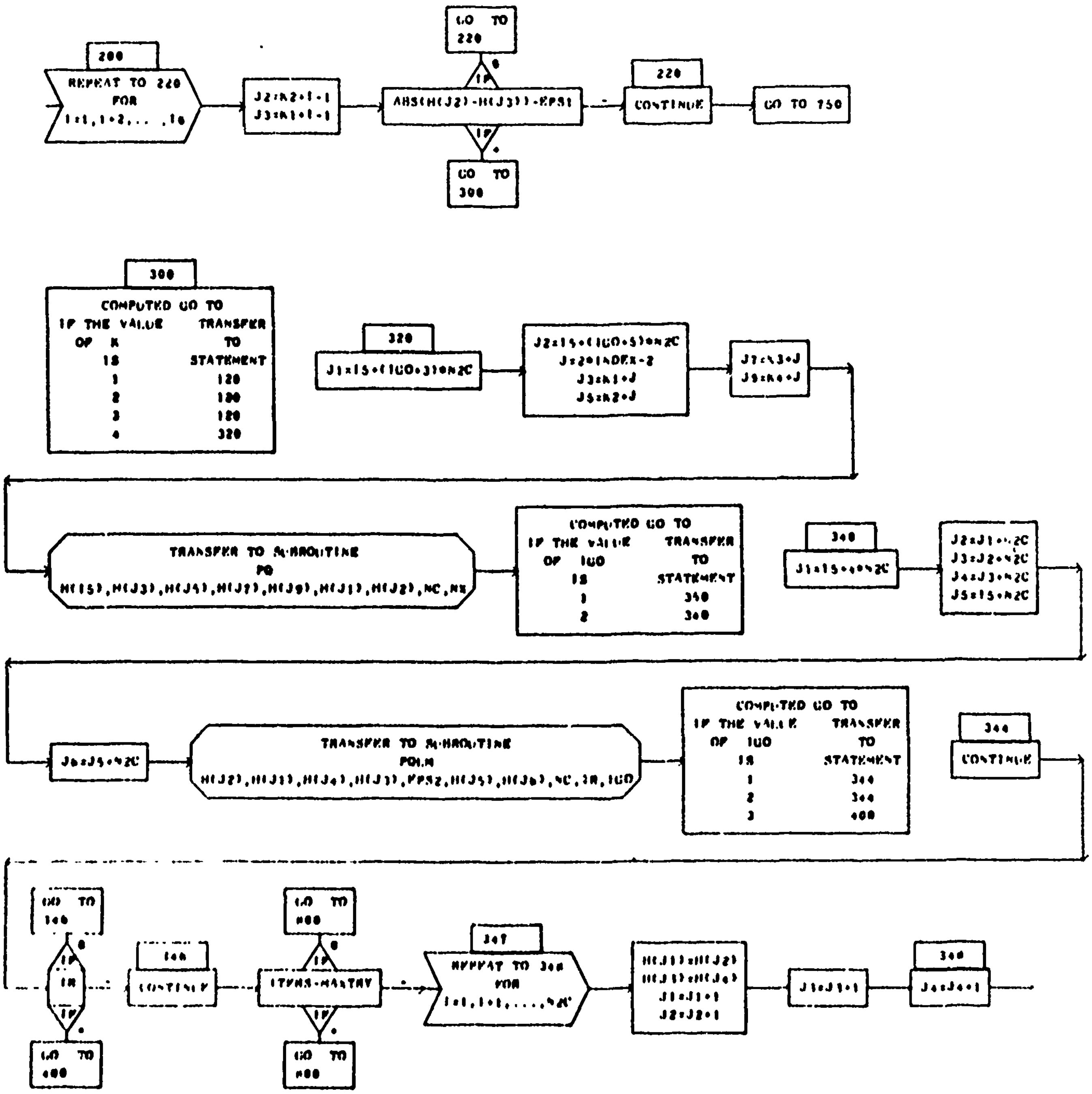
+ FOR SINGLE ROOT

NC = 1, REAL 2, COMPLEX

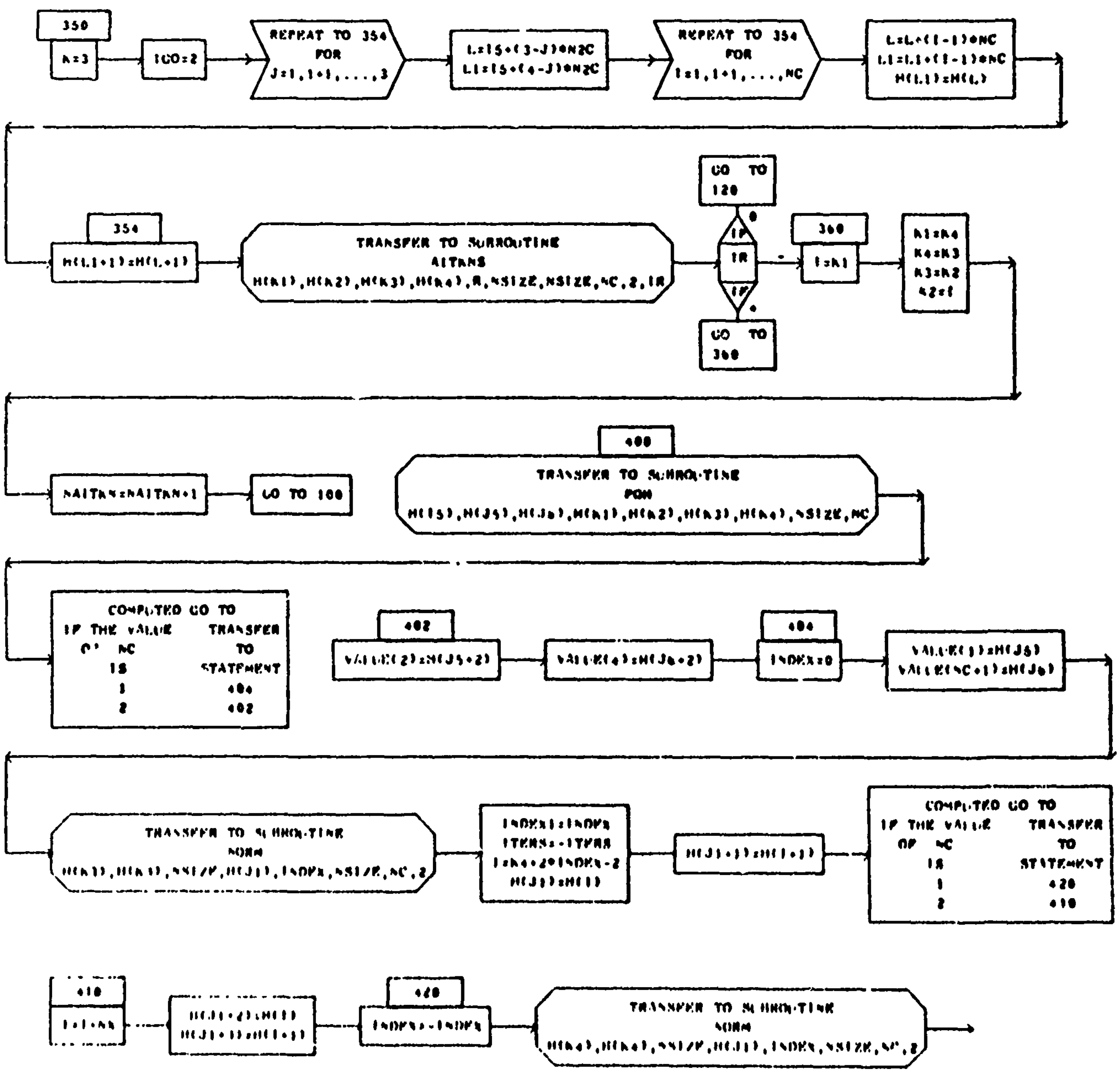
D I M E N S I O N E D V A R I A B L E S

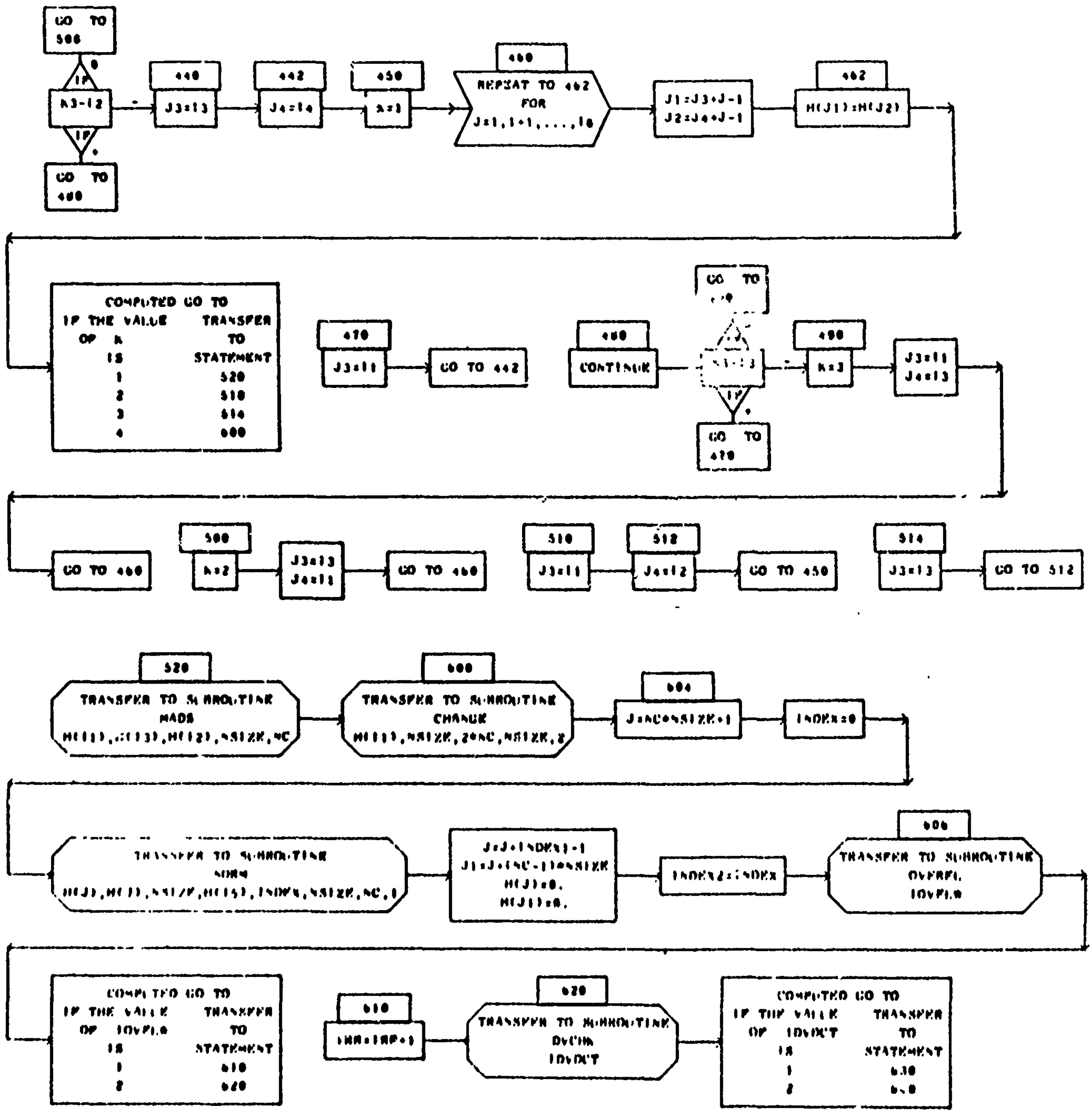
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
U	1	H	1	VALUE	1				

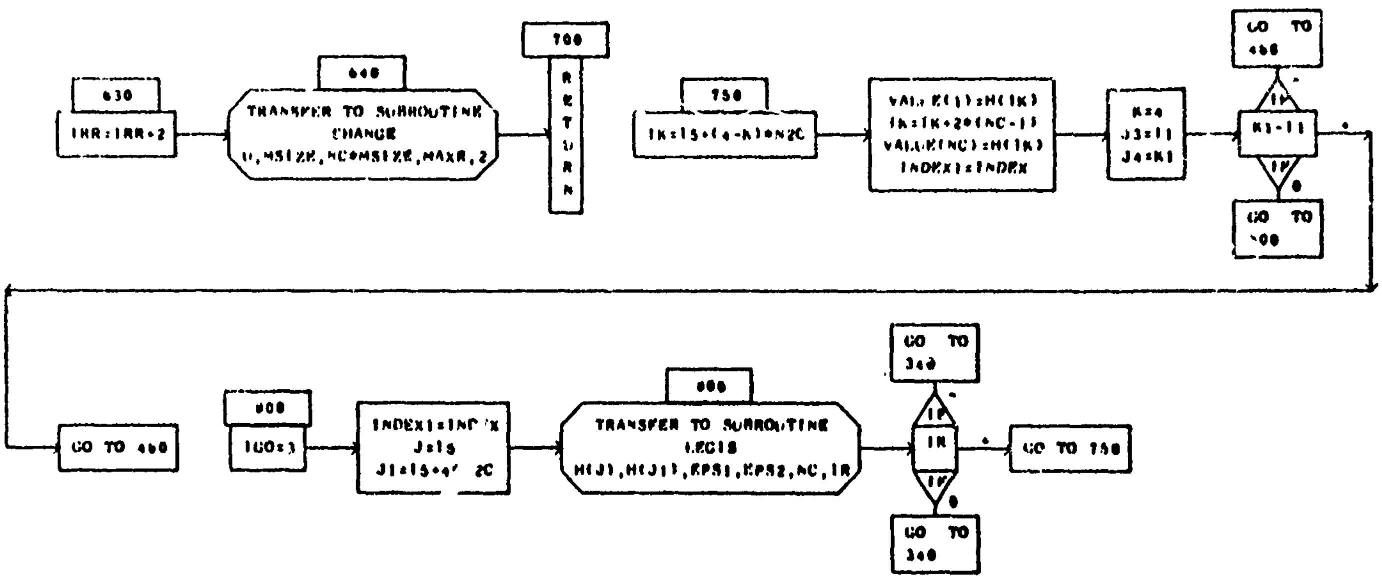

START
TRANSFER TO SUBROUTINE OVERFL IOVFL
TRANSFER TO SUBROUTINE DVCHK IOVDCT
IRR=0
TRANSFER TO SUBROUTINE CHANGE
TRANSFER TO SUBROUTINE CHANGE H,NSIZE,NC,NSIZE,1
100
120
ITERS=ITERS+1
GO TO 140
CONTINUE
GO TO 132
132
CONTINUE
TRANSFER TO SUBROUTINE
INDEX=0
COMPUTED GO TO
IF THE VALUE OF IOVFL IS
TRANSFER TO STATEMENT
1 150
2 160
150
IRR=IRR+1
160
COMPUTED GO TO
IF THE VALUE OF IOVDCT IS
TRANSFER TO STATEMENT
1 170
2 180
170
IRR=IRR+2
180
CONTINUE
IRR
GO TO 200


200
REPEAT TO 220 FOR
GO TO 220
IF
GO TO 300
220
CONTINUE
GO TO 750
300
COMPUTED GO TO
IF THE VALUE OF K IS
TRANSFER TO STATEMENT
320
TRANSFER TO SUBROUTINE
COMPUTED GO TO
340
CONTINUE
GO TO
GO TO 400
CONTINUE
REPEAT TO 348 FOR

SUBROUTINE CLOSES (U,H,NSIZE,MAXR,R,EPS1,EPS2,NC,IRR,MAXTRY,ITERS, PAGE 3




GO TO
440
442
450
460
REPEAT TO 462
FOR
462
COMPUTED GO TO
IF THE VALUE
TRANSFER
TO
IS
STATEMENT
1
520
2
3
4
600
470
J3=11
GO TO 442
480
CONTINUE
490
500
510
J3=11
512
GO TO 450
514
J3=13
GO TO 512
520
TRANSFER TO SUBROUTINE
600
CHANGE
604
610
620
TRANSFER TO SUBROUTINE
1
2
630


630
IRR=IRR+2
640
TRANSFER TO SUBROUTINE CHANGE
U,NSIZE,NC*NSIZE,MAXR,2
700
RETURN
750
GO TO
GO TO 460
GO TO 750

NITERS

A IS STORED IN CORE AT A.   (MAXR X NC*NP*NSIZE)

NTAPUT IS A UTILITY TAPE, FOR CHECK VECTORS IF DESIRED.

EPSP = EPSILON ONE = SINGLE PRECISION CONVERGENCE TEST NUMBER

EPDP = EPSILON TWO = DOUBLE PRECISION   ..   ..   ..

NC = 1, IF REAL   NP = 1, IF SINGLE PRECISION

2, IF COMPLEX   = 2, IF DOUBLE   ..

NGUESS = 6, IF FIRST GUESS IS TO BE A COLUMN OF ONES.

MODOUT = NO. OF MODES TO BE COMPUTED.

NAKSR = NO. TIMES AITKNS ACCELERATION WAS USED IN SINGLE PRECISION.

NAKDR = ... DOUBLE   ..   .

MAXSR = MAXIMUM ITERATIONS ALLOWED IN SINGLE PRECISION.

MAXDR = ... DOUBLE   ..   .

IRR = ERROR RETURN = 1, FOR OVERFLOW

2, FOR DIVIDE CHECK

3, FOR BOTH OVERFLOW AND DIVIDE CHECK

NSIZE = NO. OF ROWS AND COLUMNS OF A

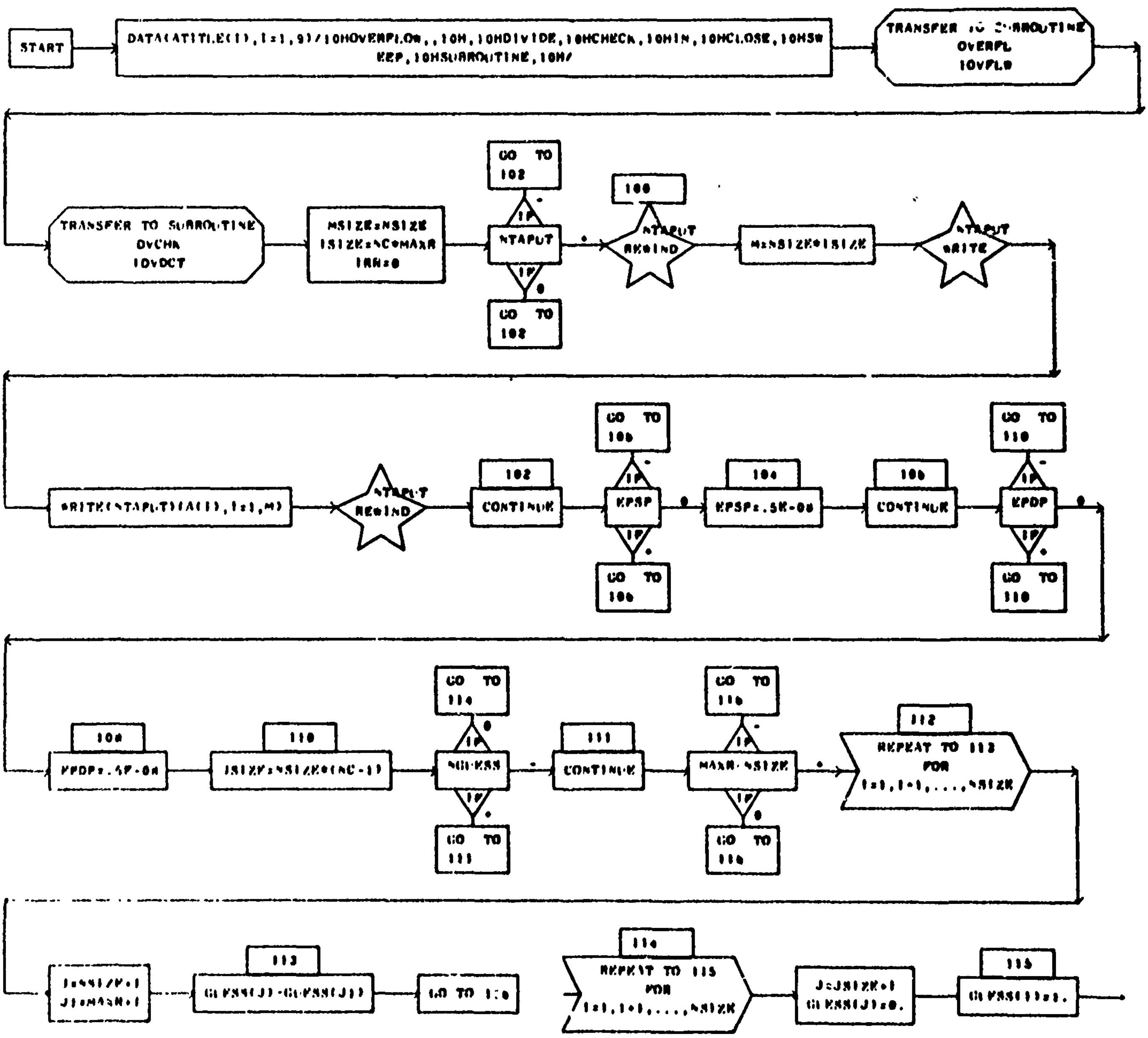
RSP = R, AITKENS ACCELERATION CONVERGENCE CONTROL FOR SINGLE PRECIS.

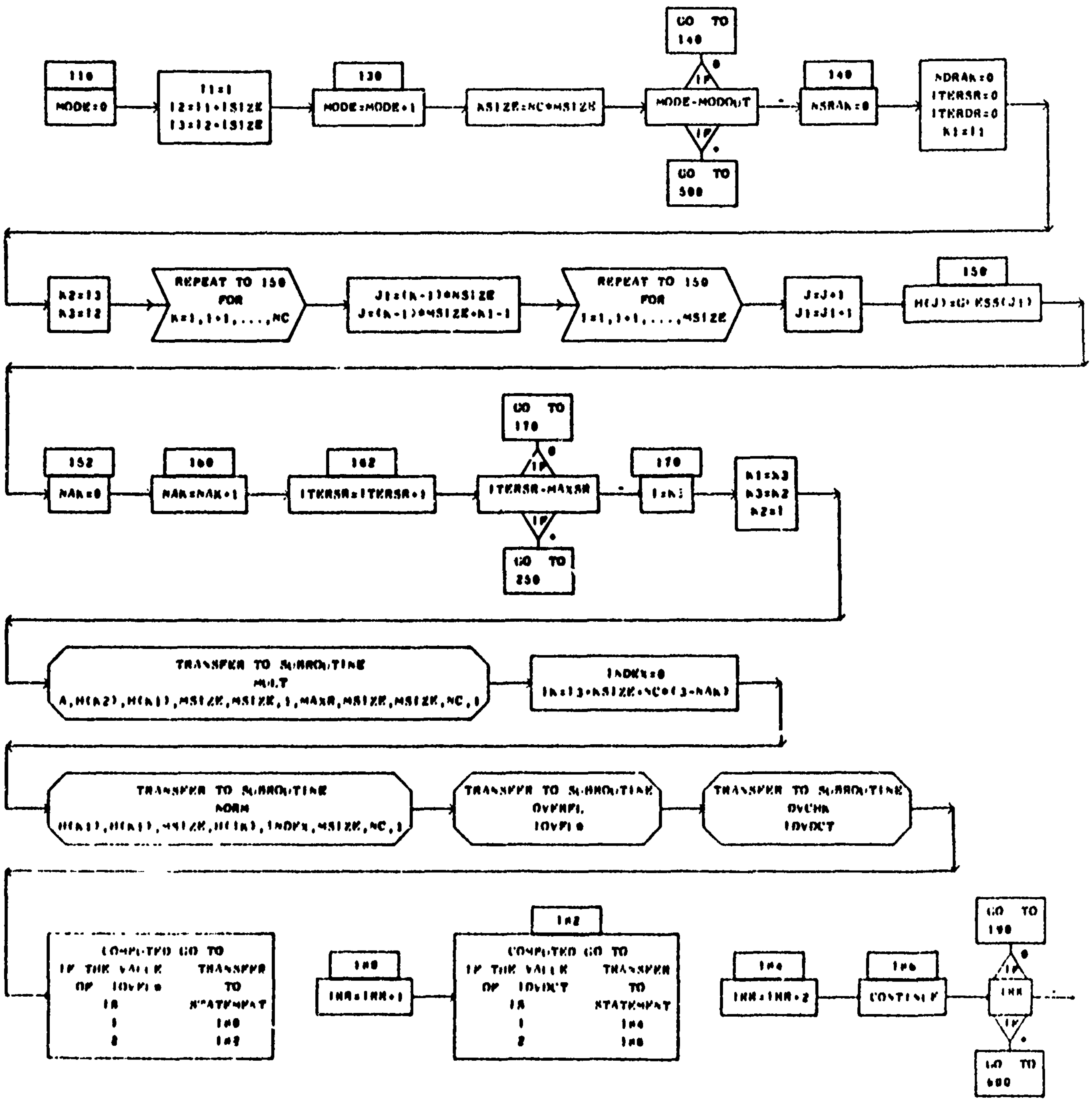
RDP = R AITKENS ACCELERATION CONVERGENCE CONTROL FOR DOUBLE PRECIS.

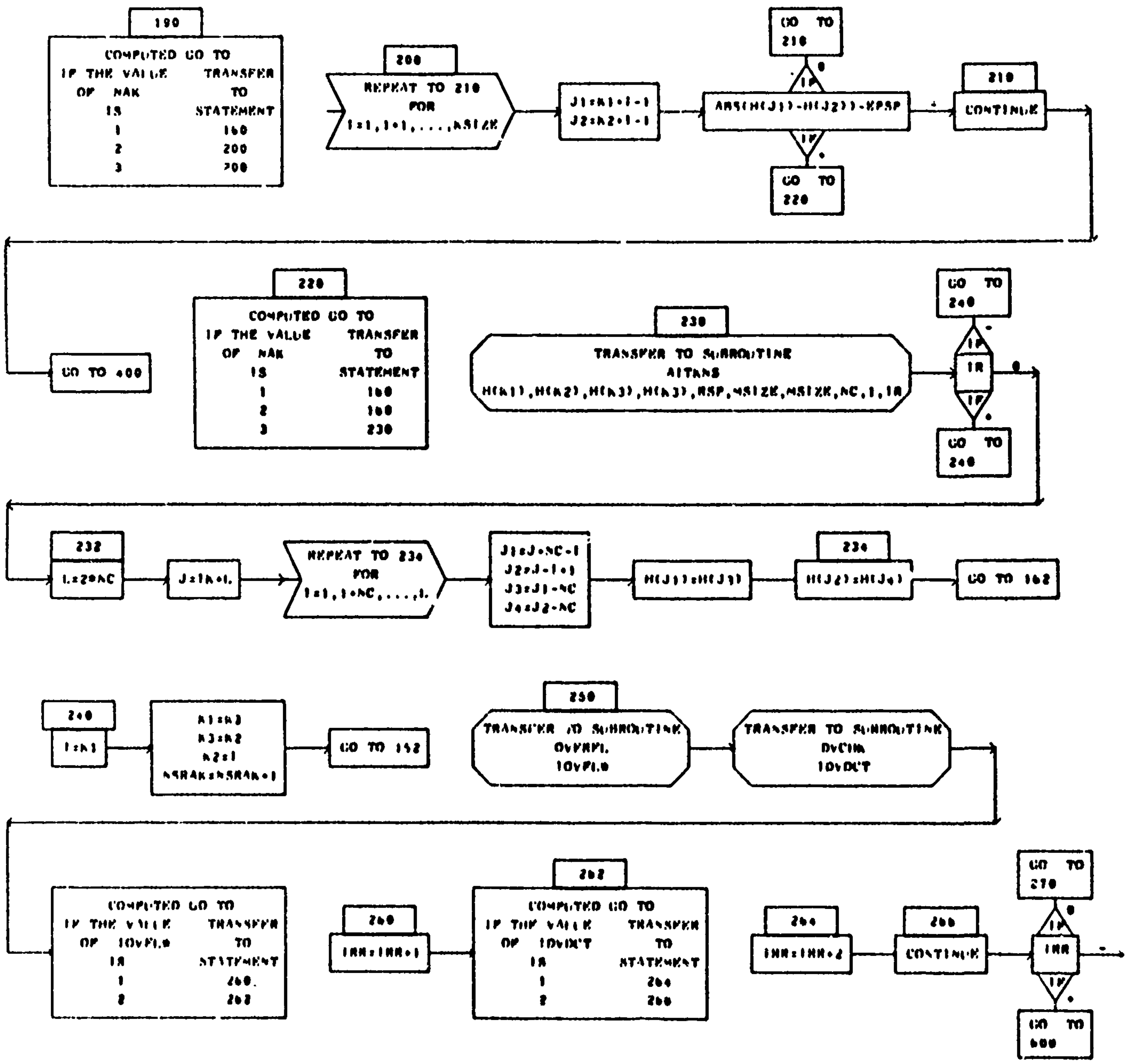
MAXR = DIMENSIONED NUMBER OF ROWS OF A AND GUESS

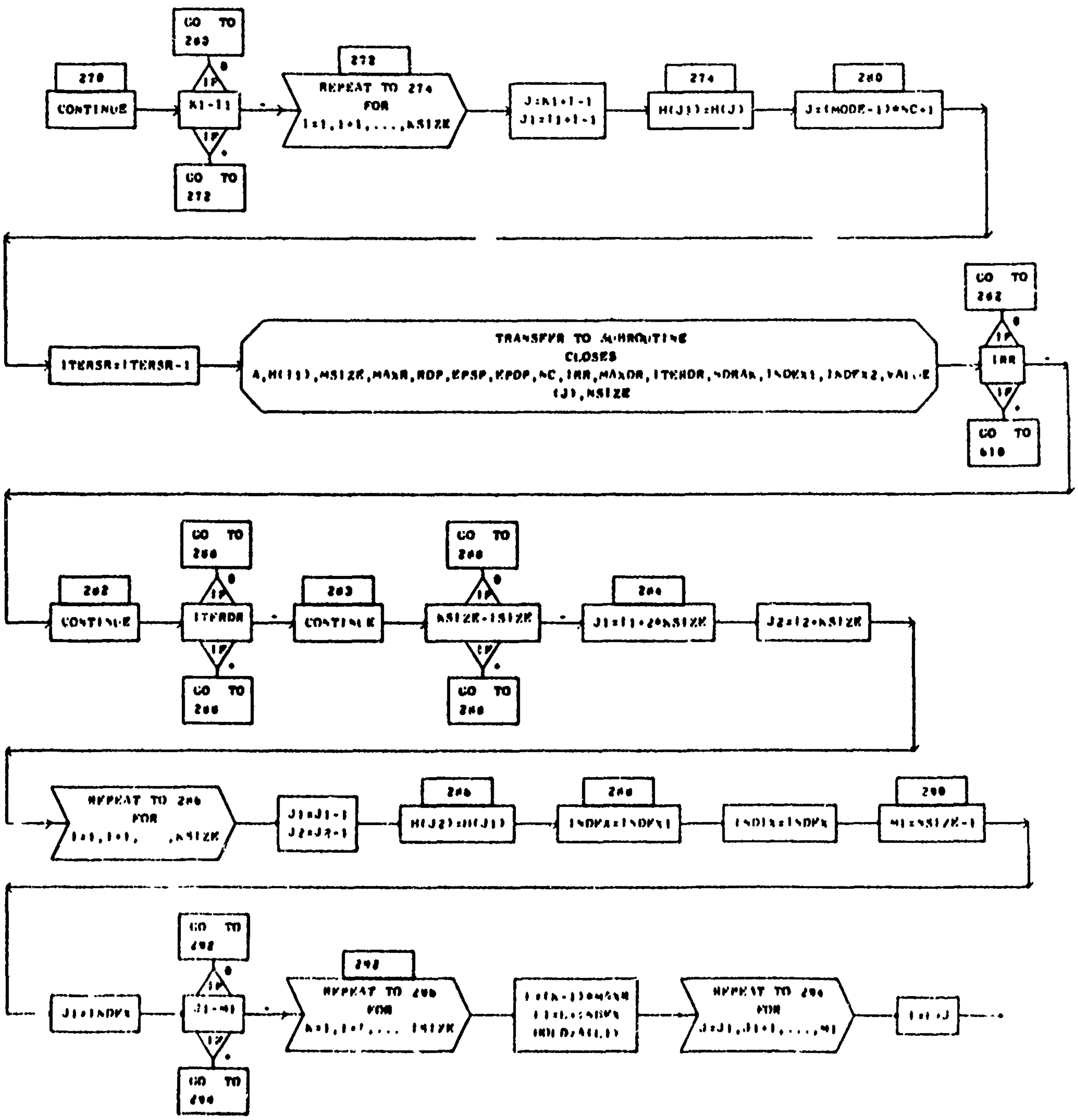
D I M E N S I O N E D   V A R I A B L E S

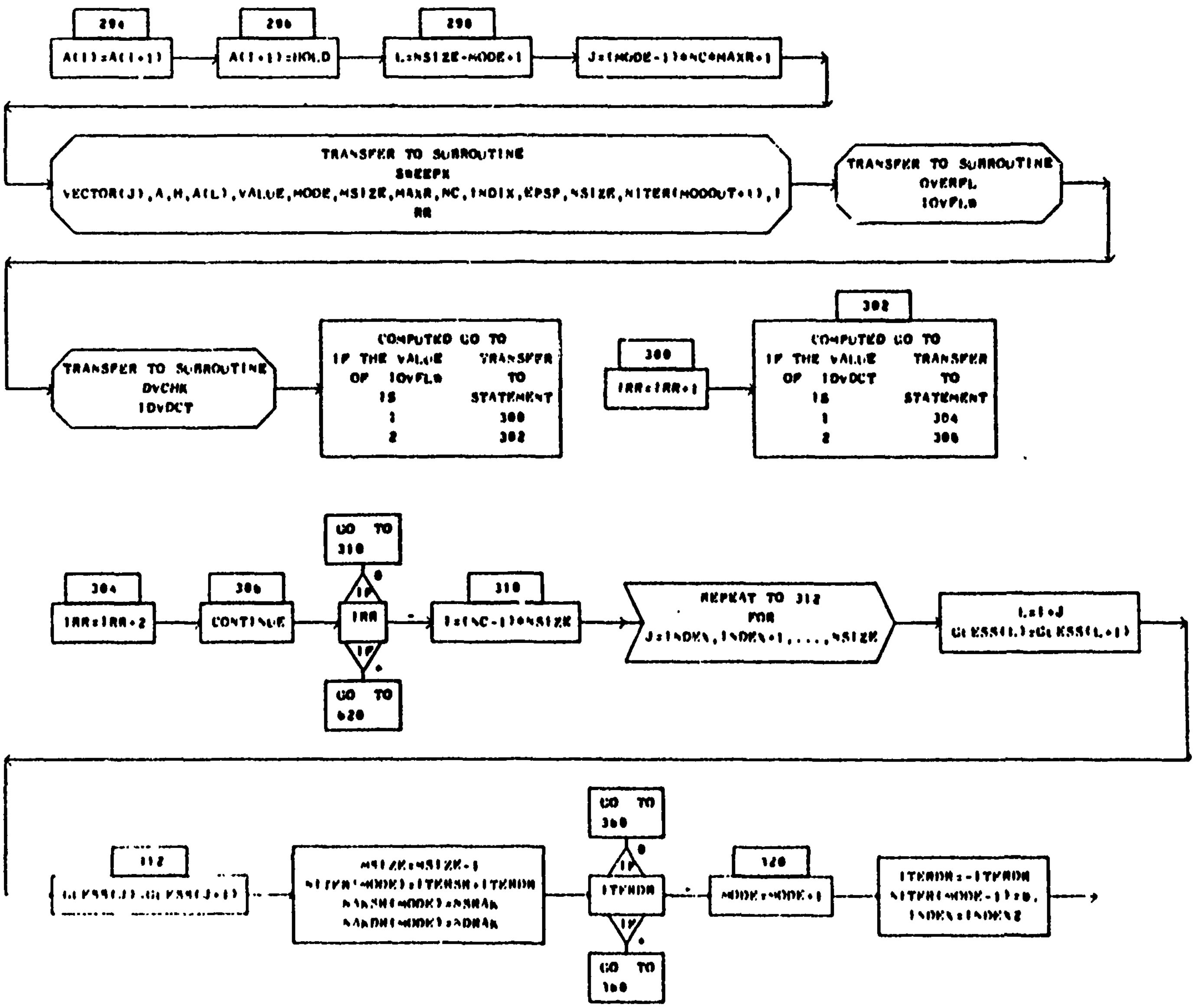
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1	GUESS	1	H	1	NITER	1	NAKSR	1
NAKDR	1	ATITLE	9	VECTOR	1	VALUE	1		

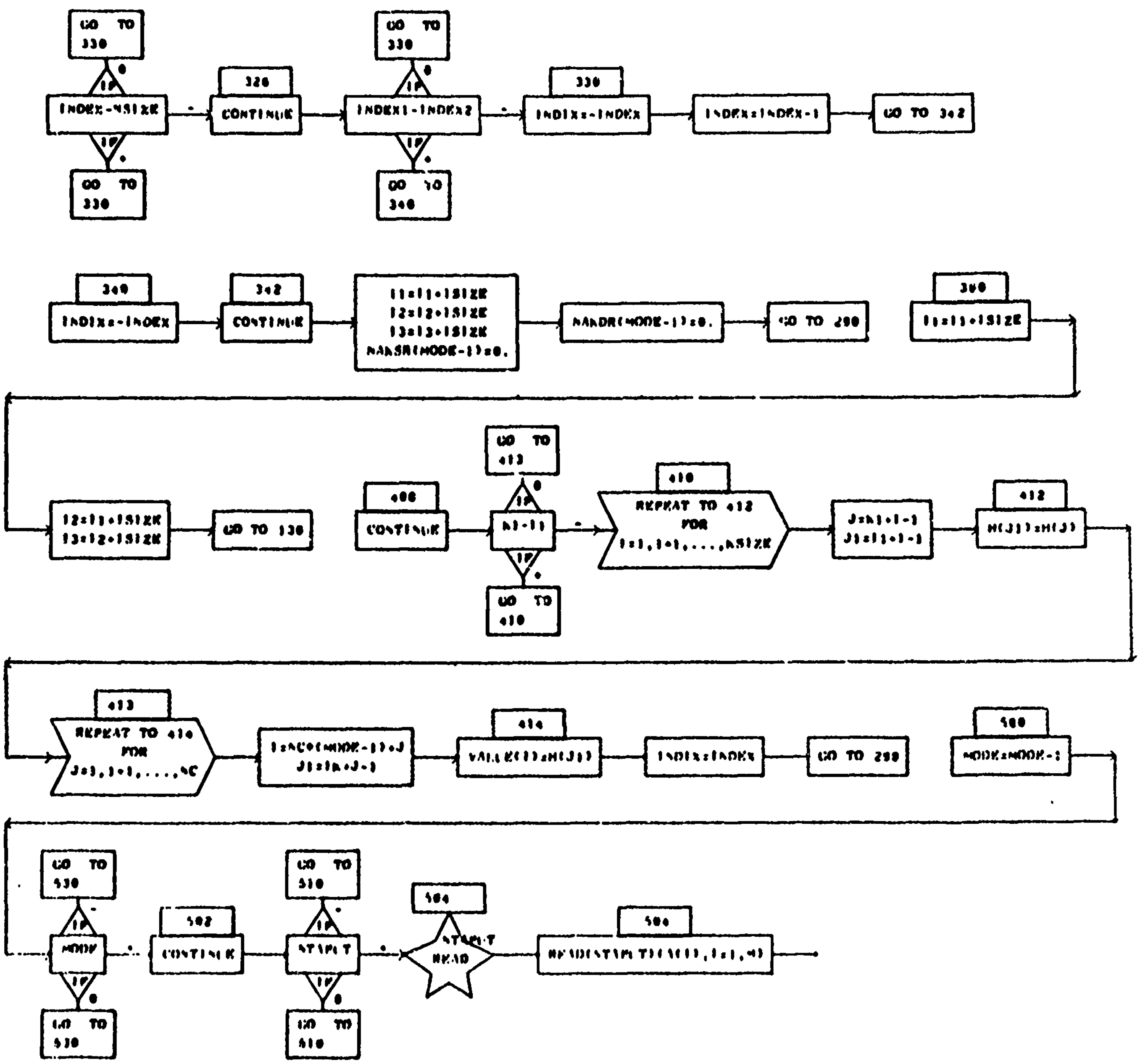

START
DATA(ATITLE(I),I=1,9)/10HOVERFLOW,,10H,10HDIVIDE,10HCHECK,10HIN,10HCLOSE,10HSW
EEP,10HSUBROUTINE,10H/
TRANSFER TO SUBROUTINE
OVERFL
IOVFLO
TRANSFER TO SUBROUTINE
DVCHK
IOVDCT
NSIZE=NSIZE
ISIZE=NC*MAXR
IRR=0
NTAPUT
REWIND
WRITE
CONTINUE
EPSP
EPDP
CONTINUE
REPEAT TO 113
FOR
REPEAT TO 115
FOR
GUESS(I)=1.

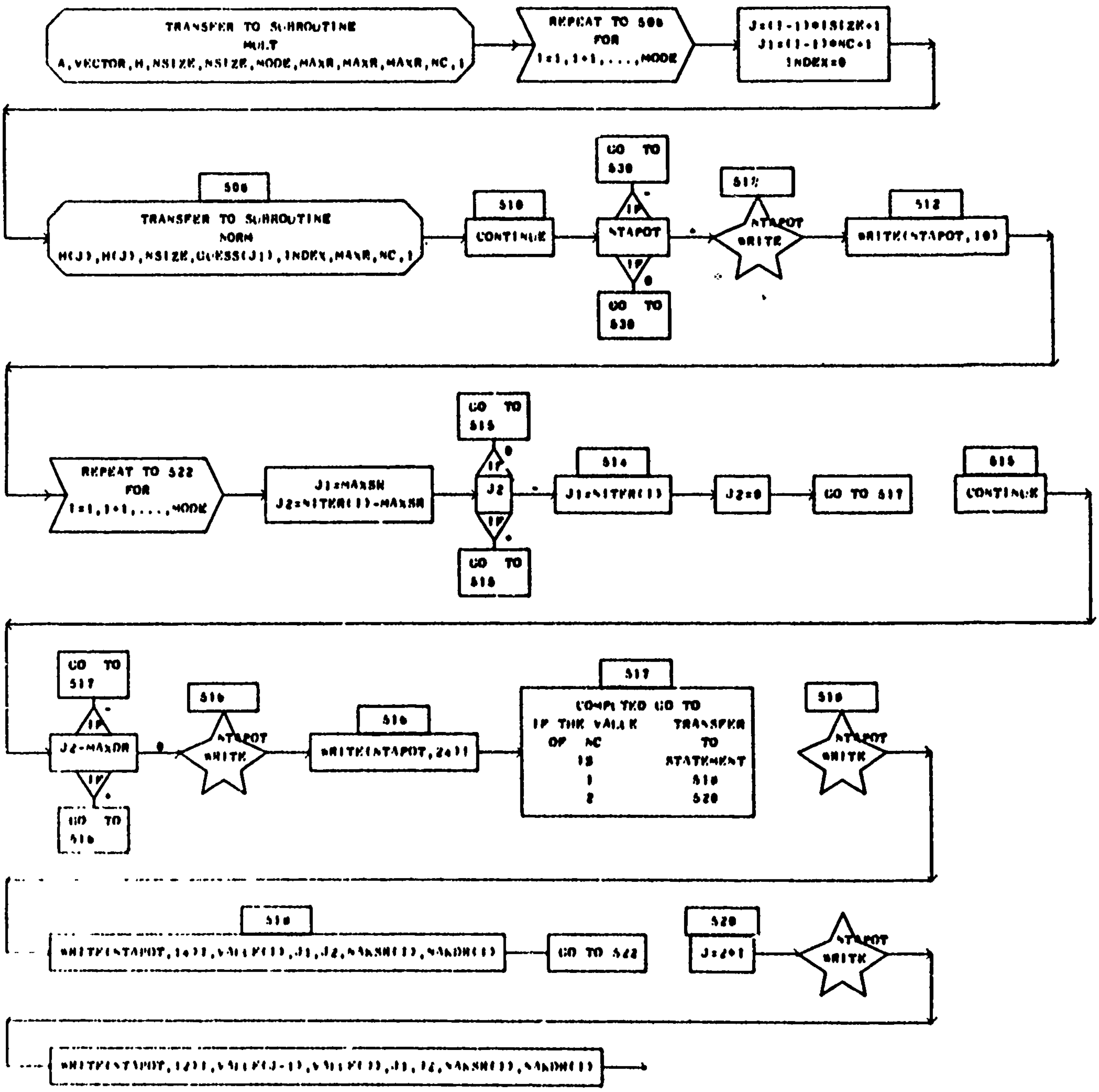

GO TO 140
110
MODE=0
11=1
12=11+1S1ZE
13=12+1S1ZE
130
MODE=MODE+1
NSIZE=NC*NSIZE
MODE-MODOUT
IF
140
NSRAK=0
NDRAK=0
ITERSR=0
ITERDR=0
K1=11
GO TO 500
K2=13
K3=12
REPEAT TO 150 FOR
J1=(K-1)*NSIZE
REPEAT TO 150 FOR
J=J+1
J1=J1+1
150
GO TO 170
152
NAK=0
160
NAK=NAK+1
162
ITERSR=ITERSR+1
ITERSR-MAXSR
170
GO TO 250
TRANSFER TO SUBROUTINE
MULT
TRANSFER TO SUBROUTINE
NORM
TRANSFER TO SUBROUTINE
TRANSFER TO SUBROUTINE
COMPUTED GO TO
IF THE VALUE
TRANSFER
TO
STATEMENT
COMPUTED GO TO
IF THE VALUE
TRANSFER
TO
STATEMENT
CONTINUE
GO TO 190
GO TO 600

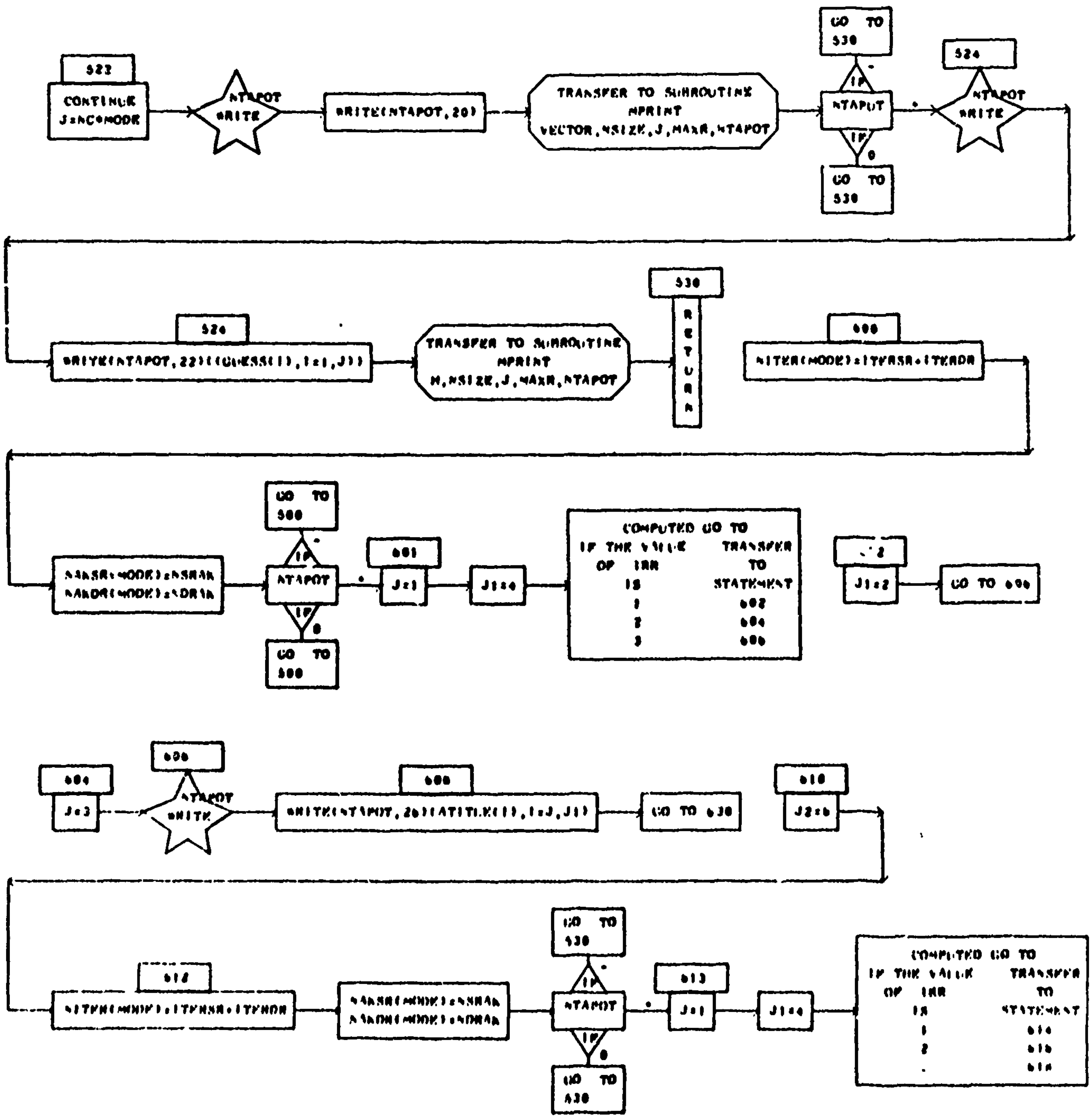

190
COMPUTED GO TO
IF THE VALUE OF NAK IS 1 2 3
TRANSFER TO STATEMENT 160 200 208
200
REPEAT TO 210 FOR I=1,1+1,...,NSIZE
J1=K1+I-1
J2=K2+I-1
ABS(H(J1)-H(J2))-EPSP
GO TO 210
GO TO 220
210
CONTINUE
GO TO 400
220
COMPUTED GO TO
IF THE VALUE OF NAK IS 1 2 3
TRANSFER TO STATEMENT 160 160 230
230
TRANSFER TO SUBROUTINE AITKNS
H(K1),H(K2),H(K3),H(K3),RSP,NSIZE,NSIZE,NC,1,IR
GO TO 240
IR
GO TO 240
232
L=2*NC
J=K+L
REPEAT TO 234 FOR I=1,1+NC,...,L
J1=J+NC-1
J2=J-1+1
J3=J1-NC
J4=J2-NC
H(J1)=H(J3)
234
H(J2)=H(J4)
GO TO 162
240
I=K1
K1=K3
K3=K2
K2=I
NSRAK=NSRAK+1
GO TO 152
250
TRANSFER TO SUBROUTINE OVERFL 10VFL0
TRANSFER TO SUBROUTINE DVCHK 10VDCT
COMPUTED GO TO
IF THE VALUE OF IOVFLO IS 1 2
TRANSFER TO STATEMENT 260 262
260
IRR=IRR+1
262
COMPUTED GO TO
IF THE VALUE OF IOVDCT IS 1 2
TRANSFER TO STATEMENT 264 266
264
IRR=IRR+2
266
CONTINUE
GO TO 270
IRR
GO TO 600

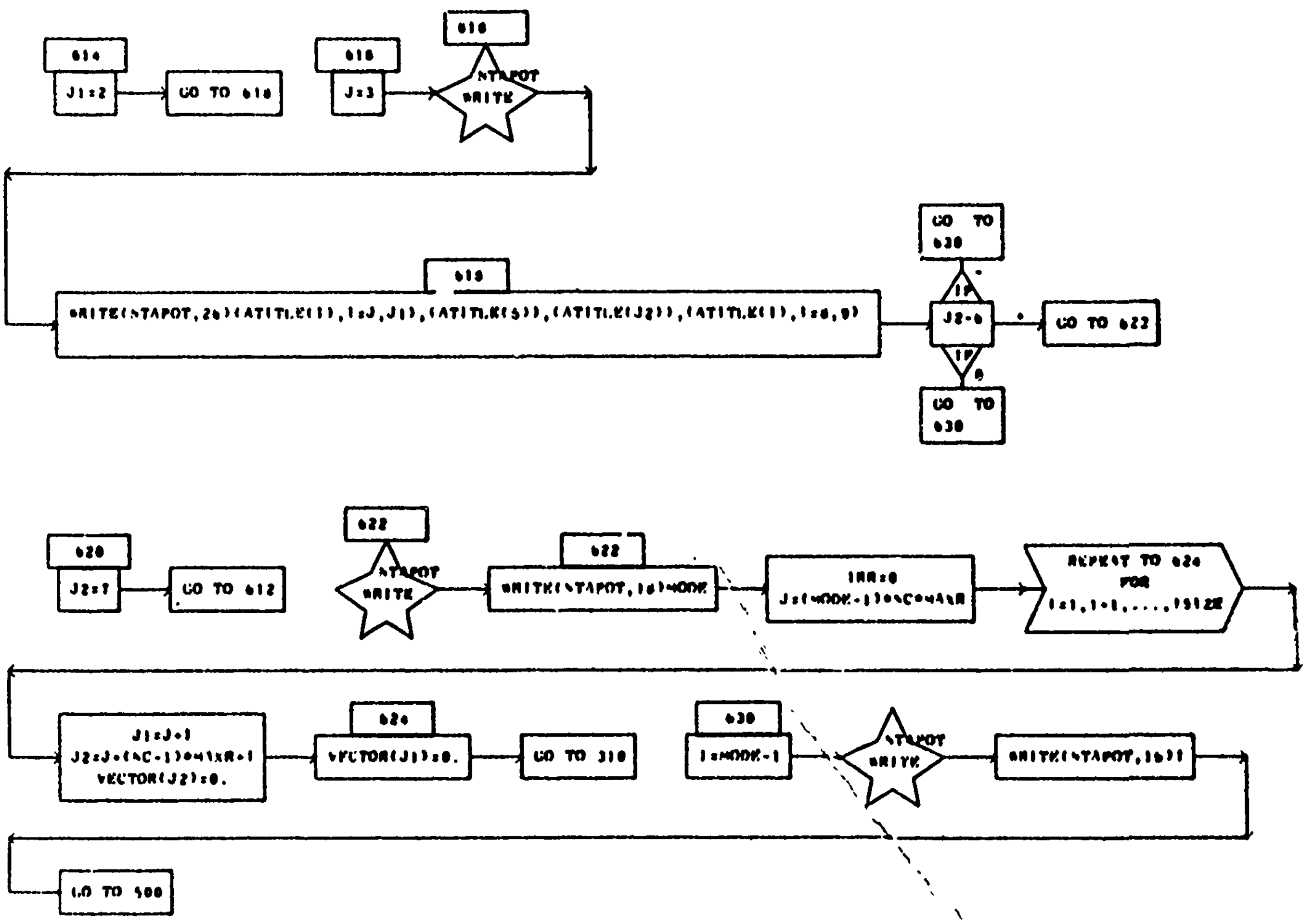



294
A(I)=A(I+1)
296
A(I+1)=HOLD
298
L=NSIZE-MODE+1
J=(MODE-1)*NC*MAXR+1
TRANSFER TO SUBROUTINE
SWEEPX
VECTOR(J),A,H,A(L),VALUE,MODE,NSIZE,MAXR,NC,INDIX,EPSP,NSIZE,NITER(MODOUT+1),I
RR
TRANSFER TO SUBROUTINE
OVERFL
IOVFLW
TRANSFER TO SUBROUTINE
DVCHK
IDVDCT
COMPUTED GO TO
IF THE VALUE OF IOVFLW IS
TRANSFER TO STATEMENT
1 300
2 302
300
IRR=IRR+1
302
COMPUTED GO TO
IF THE VALUE OF IDVDCT IS
TRANSFER TO STATEMENT
1 304
2 306
304
IRR=IRR+2
306
CONTINUE
IRR
GO TO 310
GO TO 620
310
REPEAT TO 312
FOR
ITERDR
GO TO 360
GO TO 160
320
MODE=MODE+1


GO TO 330
INDEX-NSIZE
320
CONTINUE
INDEX1-INDEX2
330
INDEX1-INDEX2
GO TO 342
340
CONTINUE
GO TO 290
GO TO 130
CONTINUE
REPEAT TO 412 FOR
GO TO 413
GO TO 410
REPEAT TO 414 FOR
GO TO 290
CONTINUE
START READ


TRANSFER TO SUBROUTINE
MULT
A,VECTOR,H,NSIZE,NSIZE,MODE,MAXR,MAXR,MAXR,NC,1
REPEAT TO 506
FOR
I=1,I+1,...,MODE
J=(I-1)*NSIZE+1
J1=(I-1)*NC+1
INDEX=0
506
TRANSFER TO SUBROUTINE
NORM
H(J),H(J),NSIZE,GUESS(J1),INDEX,MAXR,NC,1
510
CONTINUE
GO TO
530
IF
NTAPOT
GO TO
530
511
NTAPOT
WRITE
512
WRITE(NTAPOT,10)
REPEAT TO 522
FOR
I=1,I+1,...,MODE
J1=MAXSR
J2=NITER(I)-MAXSR
GO TO
515
IF
J2
GO TO
515
514
J1=NITER(I)
J2=0
GO TO 517
515
CONTINUE
GO TO
517
IF
J2-MAXDR
GO TO
516
516
NTAPOT
WRITE
516
WRITE(NTAPOT,24)
517
COMPUTED GO TO
IF THE VALUE   TRANSFER
OF NC   TO
IS   STATEMENT
1   518
2   520
518
NTAPOT
WRITE
518
WRITE(NTAPOT,14)(VALUE(I),J1,J2,NAKSR(I),NAKDR(I)
GO TO 522
520
J=2*I
NTAPOT
WRITE
WRITE(NTAPOT,12)(VALUE(J-1),VALUE(J),J1,J2,NAKSR(I),NAKDR(I)


522
CONTINUE
J=NC*MODE
NTAPOT
WRITE
WRITE(NTAPOT,20)
TRANSFER TO SUBROUTINE
MPRINT
VECTOR,NSIZE,J,MAXR,NTAPOT
GO TO
530
IF
NTAPOT
IF
0
GO TO
530
524
NTAPOT
WRITE
526
WRITE(NTAPOT,22)((GUESS(I),I=1,J))
TRANSFER TO SUBROUTINE
MPRINT
H,NSIZE,J,MAXR,NTAPOT
530
RETURN
600
COMPUTED GO TO
IF THE VALUE
OF IRR
IS
TRANSFER
TO
STATEMENT
GO TO
600
601
J=1
GO TO
600
604
J=3
606
NTAPOT
WRITE
608
GO TO 630
610
GO TO
630
NTAPOT
613
J=1
GO TO
630
COMPUTED GO TO
IF THE VALUE
OF IRR
IS
TRANSFER
TO
STATEMENT


614
J1=2
GO TO 618
616
J=3
618
NTAPOT
WRITE
619
GO TO
630
J2-6
GO TO 622
GO TO
630
620
J2=7
GO TO 612
622
NTAPOT
WRITE
622
IRR=0
REPEAT TO 624
FOR
624
VECTOR(J1)=0.
GO TO 310
630
NTAPOT
WRITE
GO TO 500

SWEEPS

S W E E P S   S U B R O U T I N E

COMPUTES TRUE MODE AND SWEEPS IT FROM THE MATRIX. (REAL OR COMPLEX)

HTRUE = TRUE MODAL COLUMNS, AS COMPUTED. U = DYNAMIC MATRIX.

H = SERIES OF MODIFIED MODAL COLUMNS.F = COLUMN OF EIGENVALUES.

US = SERIES OF MODIFIED MODAL ROWS OF U.

MODE = MODE NOW BEING COMPUTED.   N = SIZE

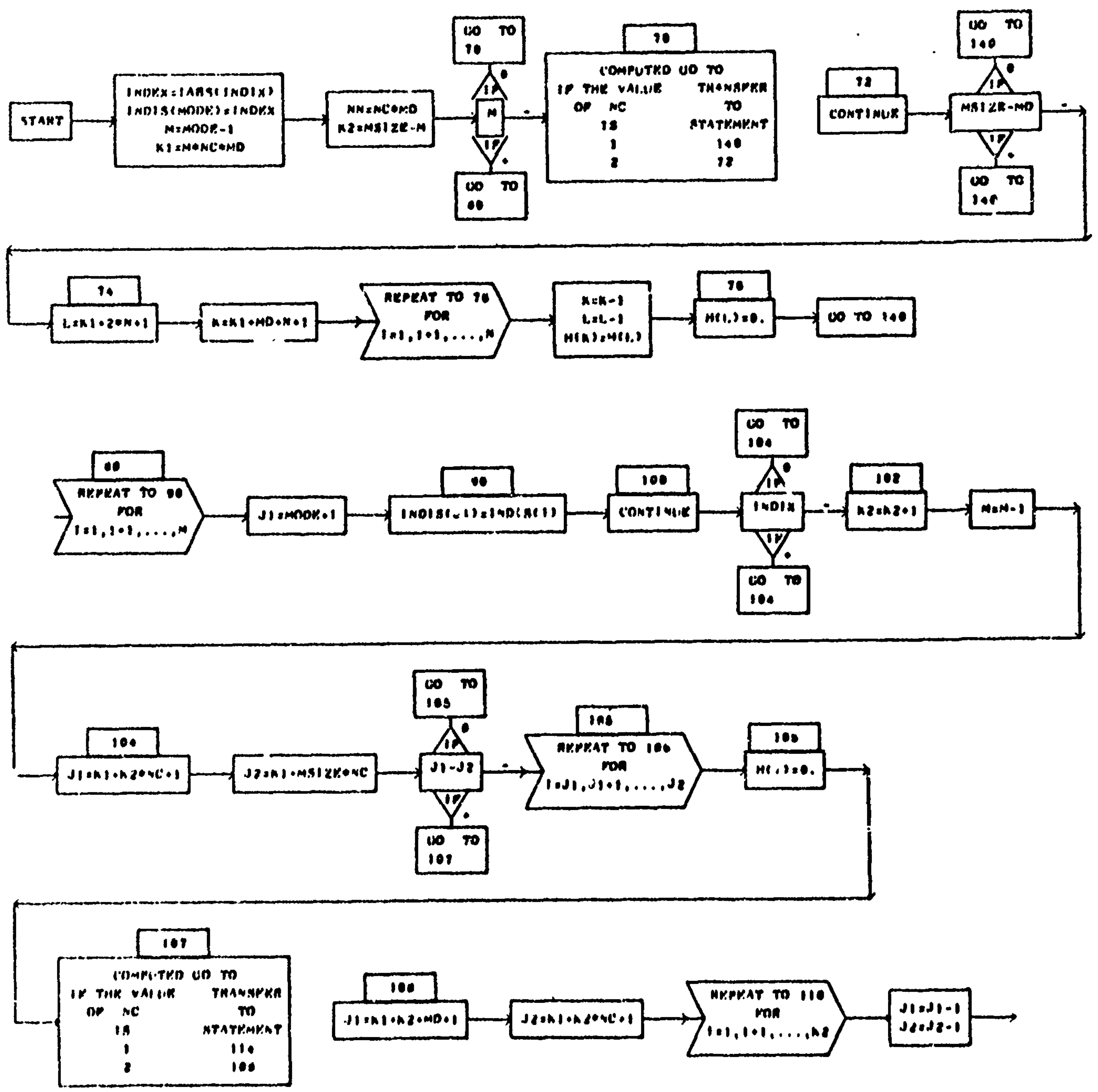
ND = DIMENSIONED NUMBER OF ROWS OF U,US,H,HTRUE

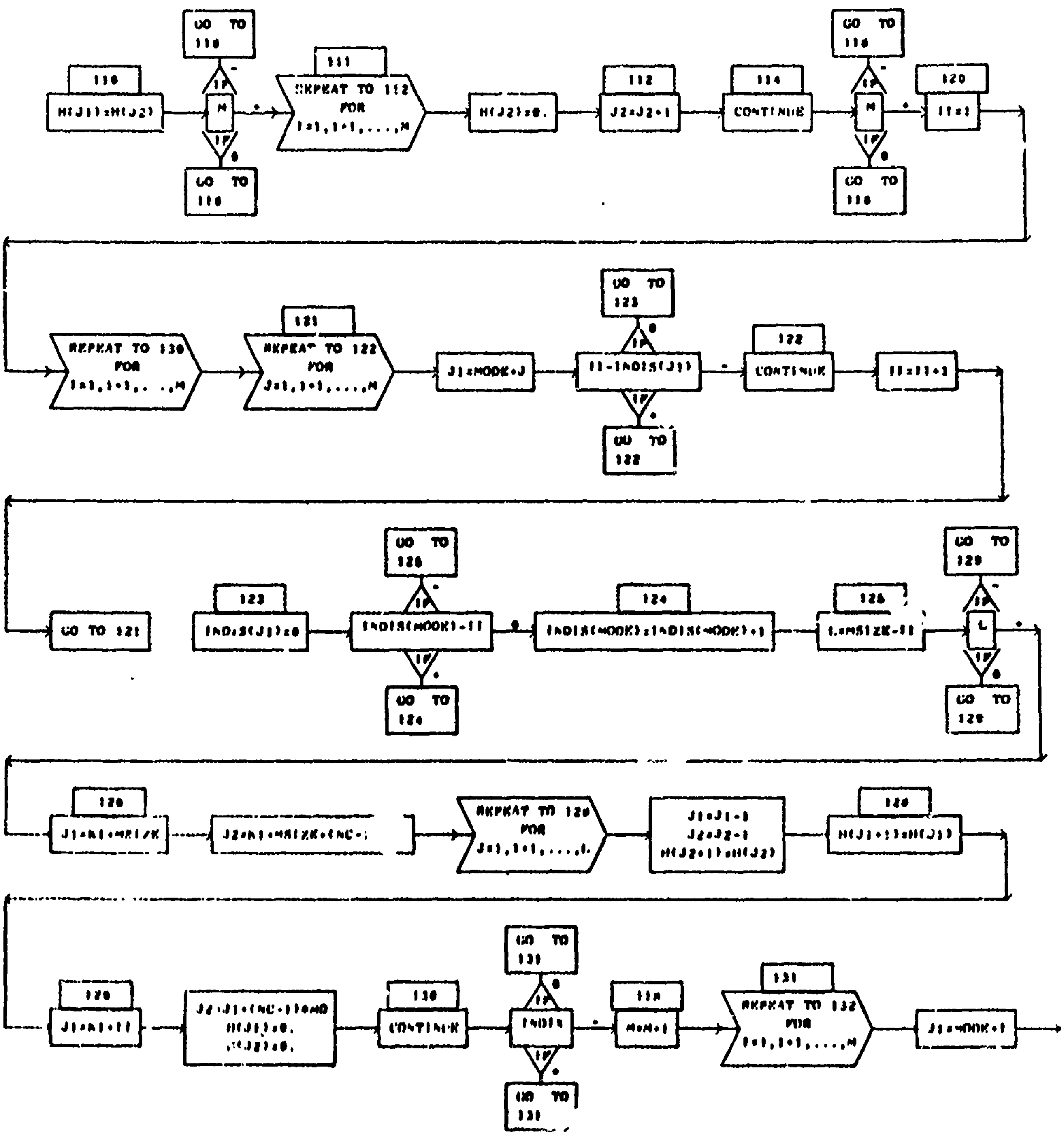
NN = 1 IF PROBLEM IS REAL.

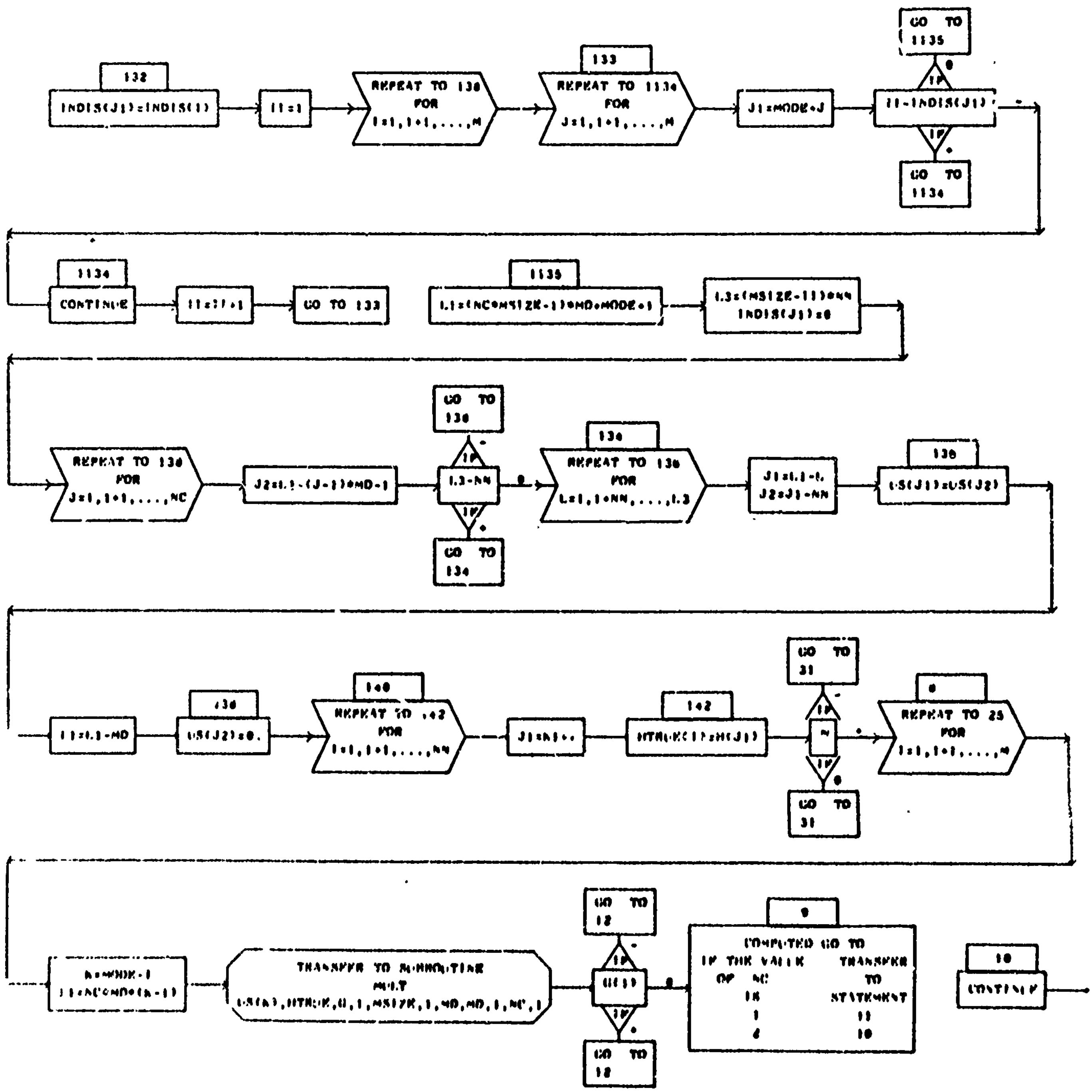
= 2 IF PROBLEM IS COMPLEX.

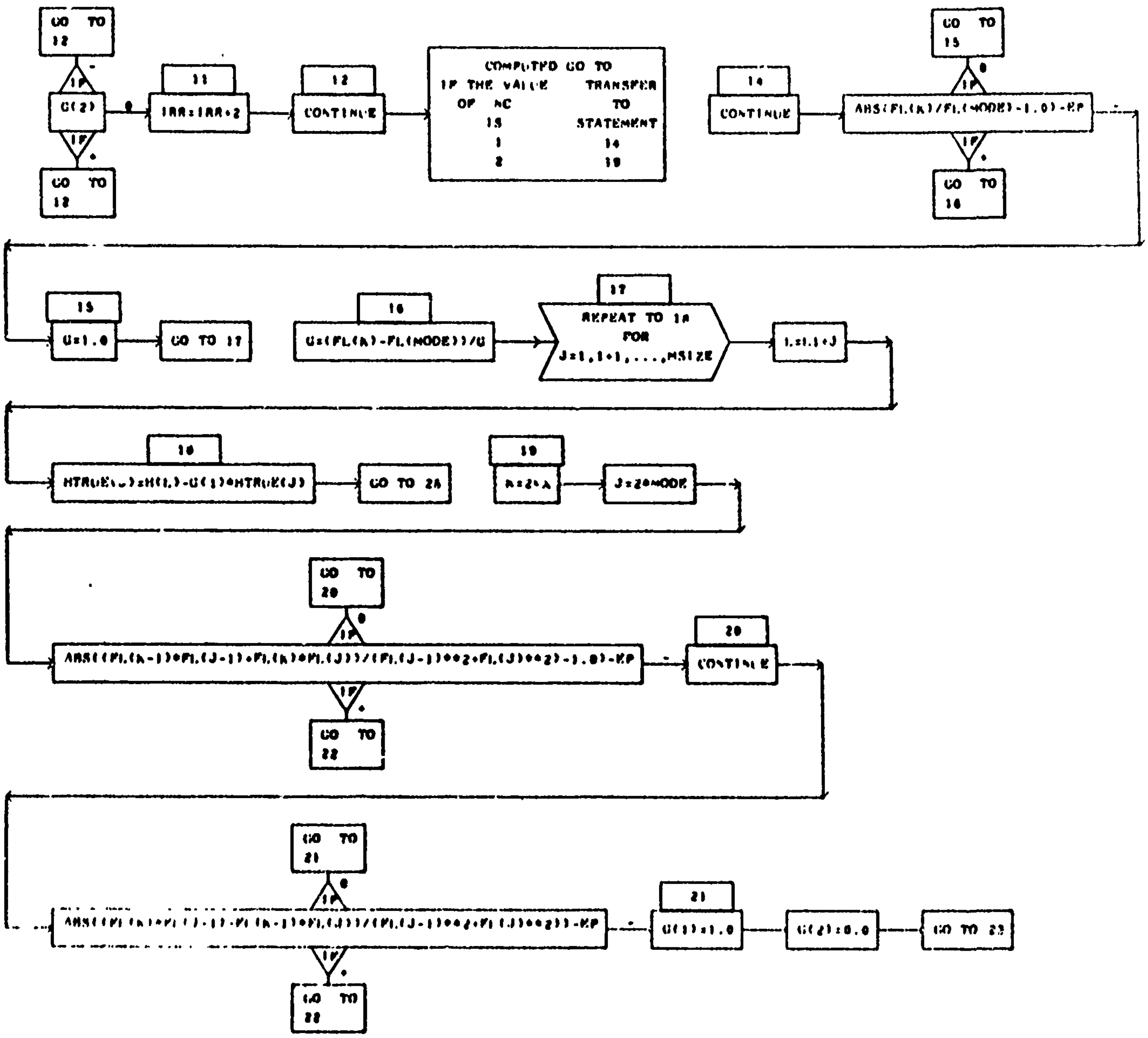
D I M E N S I O N E D   V A R I A B L E S

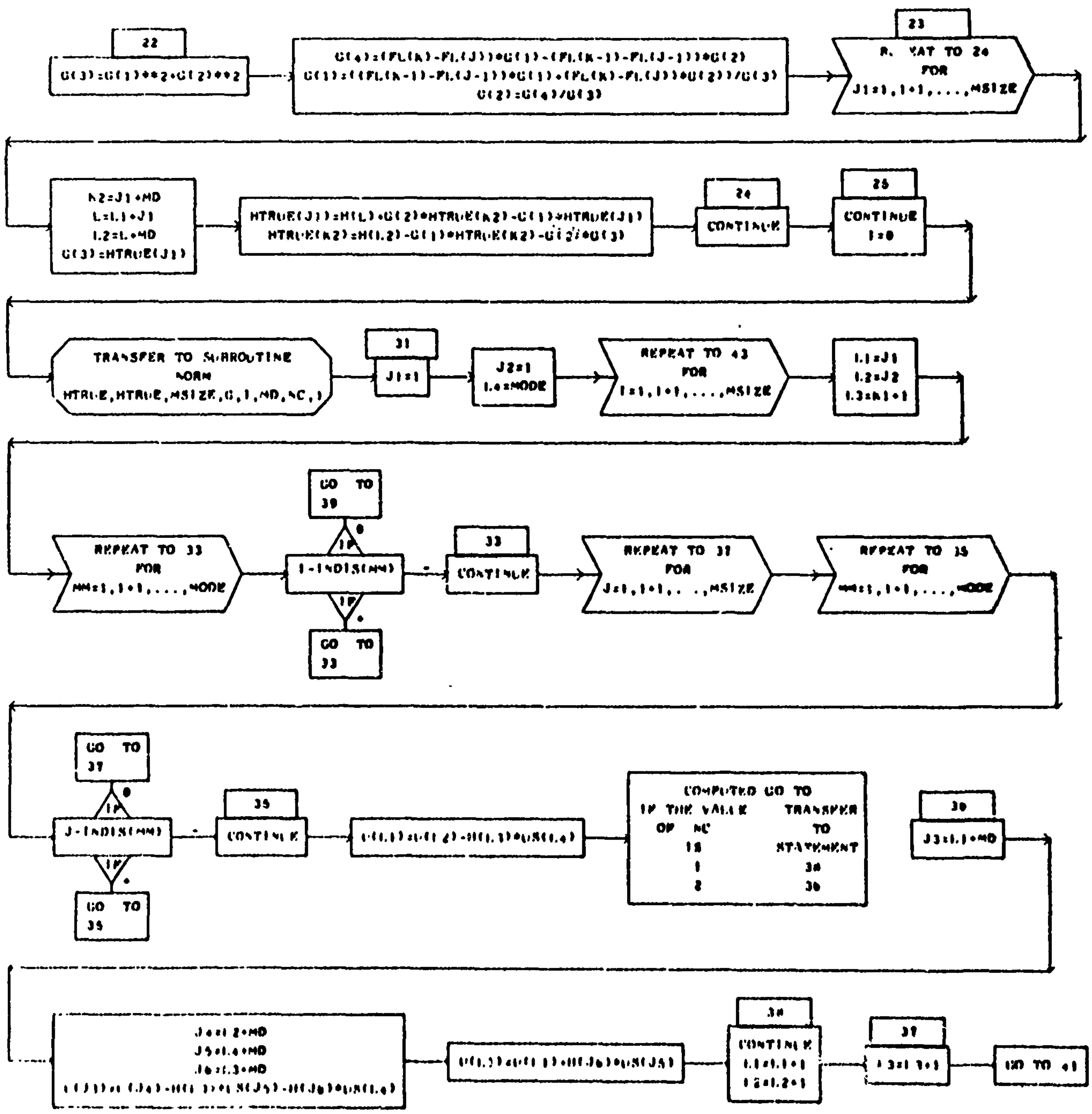
SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
H	1	US	1	U	1	HTRUE	1	F	1
G	4	INDIS	1						

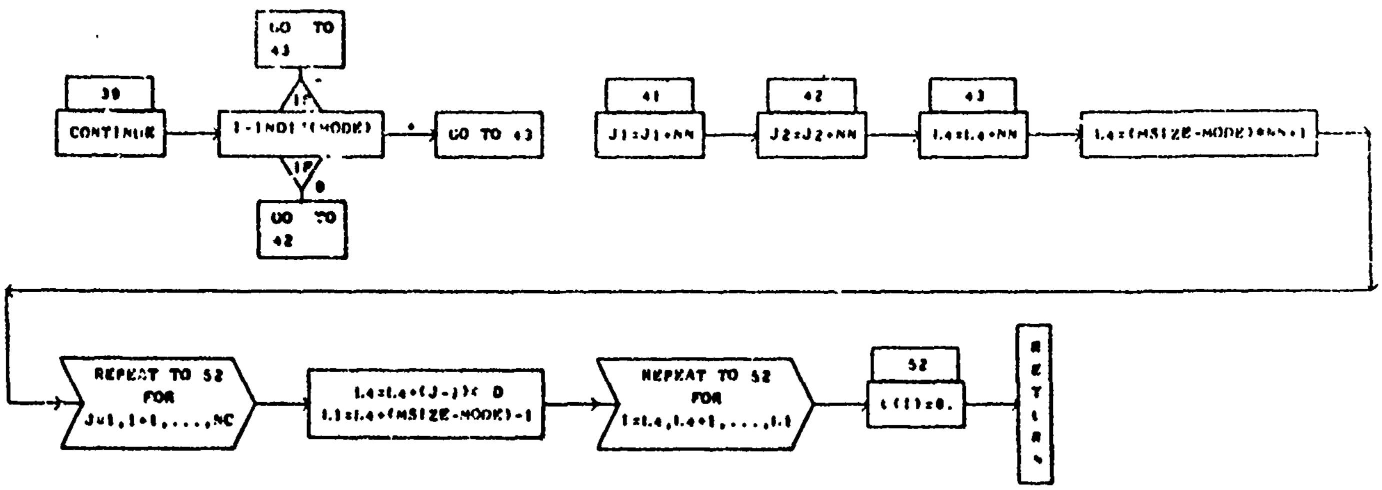

START
INDEX=IABS(INDIX)
INDIS(MODE)=INDEX
M=MODE-1
K1=M*NC*MD
NN=NC*MD
K2=MSIZE-M
GO TO 70
IF
M
IF
GO TO 80
70
COMPUTED GO TO
IF THE VALUE OF NC IS
1
2
TRANSFER TO STATEMENT
140
72
72
CONTINUE
GO TO 140
MSIZE-MD
GO TO 140
74
L=K1+2*N+1
K=K1+MD*N+1
REPEAT TO 76 FOR I=1,I+1,...,N
K=K-1
L=L-1
H(K)=H(L)
76
H(L)=0.
GO TO 140
80
REPEAT TO 90 FOR I=1,I+1,...,M
J1=MODE+1
90
INDIS(J1)=IND(K(I))
100
CONTINUE
GO TO 104
INDIX
GO TO 104
102
K2=K2+1
M=M-1
104
J1=K1+K2*NC+1
J2=K1+MSIZE*NC
GO TO 105
J1-J2
GO TO 107
105
REPEAT TO 106 FOR I=J1,J1+1,...,J2
106
H(I)=0.
107
COMPUTED GO TO
IF THE VALUE OF NC IS
1
2
TRANSFER TO STATEMENT
110
108
108
J1=K1+K2*MD+1
J2=K1+K2*NC+1
REPEAT TO 110 FOR I=1,I+1,...,K2
J1=J1-1
J2=J2-1




132
INDIS(J1)=INDIS(I)
I1=1
REPEAT TO 138
FOR
I=1,1+1,...,M
133
REPEAT TO 1134
FOR
J=1,1+1,...,M
J1=MODE+J
I1-INDIS(J1)
GO TO
1135
GO TO
1134
1134
CONTINUE
I1=I1+1
GO TO 133
1135
REPEAT TO 138
FOR
J=1,1+1,...,NC
GO TO
136
GO TO
134
134
REPEAT TO 136
FOR
136
REPEAT TO 142
FOR
I=1,1+1,...,NN
142
HTRUE(I)=H(J1)
N
GO TO
31
GO TO
31
REPEAT TO 25
FOR
I=1,1+1,...,M
TRANSFER TO SUBROUTINE
MULT
GO TO
12
GO TO
12
9
COMPUTED GO TO
IF THE VALUE
TRANSFER
OF
TO
IS
STATEMENT
1
11
4
10
10
CONTINUE


GO TO 12
IF
G(2)
IRR=IRR+2
CONTINUE
COMPUTED GO TO
IF THE VALUE OF NC IS 1 2
TRANSFER TO STATEMENT 14 19
CONTINUE
GO TO 15
ABS(FL(K)/FL(MODE)-1.0)-EP
GO TO 16
U=1.0
GO TO 17
U=(FL(K)-FL(MODE))/U
REPEAT TO 18 FOR J=1,1+1,...,MSIZE
GO TO 28
GO TO 20
CONTINUE
GO TO 22
GO TO 21
G(2)=0.0
GO TO 23



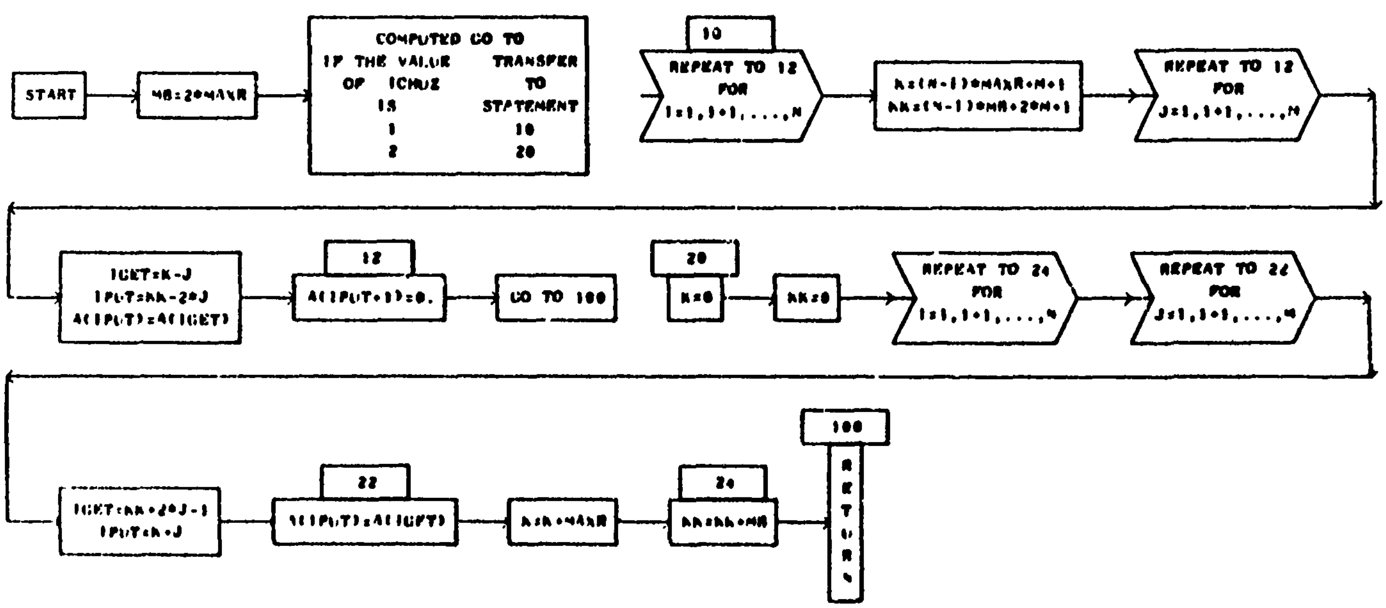

GO TO
43
39
CONTINUE
IF
GO TO 43
41
J1=J1+NN
42
J2=J2+NN
43
NN
GO TO
42
REPEAT TO 52
FOR
REPEAT TO 52
FOR
52
R
E
T
U
R
N

CHANGES

DIMENSIONED VARIABLES

SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES	SYMBOL	STORAGES
A	1								

SUBROUTINE CHANGE (A,M,N,MAXR,ICHUZ) PAGE 1



SECTION 10

NOMENCLATURE

NOMENCLATURE

a	Element of flexibility matrix, in./lb
a_R	Generalized amplitude coefficient of rigid-body modal series, in. or rad
b_r	Reference semichord, ft
C_h	Element of oscillatory aerodynamic influence coefficient matrix, dimensionless
F	Control point force, lb
g	Structural damping coefficient, dimensionless
h_o	Control point deflection due to rigid-body motion, in.
h_R	Element in rigid-body modal matrix, in. or dimensionless (see Section II)
h_1	Control point deflection, in.
K	Flexibility matrix normalizing constant, dimensionless
k_r	Reference reduced frequency, dimensionless
M	Element of mass matrix, lb.
$\bar{M}$	Element of complex mass matrix (includes aerodynamic effects), lb
m	Element of generalized mass matrix, lb., in.-lb, or lb-in^2.
$\bar{m}$	Element of sum of generalized mass and aerodynamic matrices, lb, in.-lb, or lb-in^2.
Q	Element of generalized aerodynamic force matrix, lb, in.-lb, or lb-in^2.
R	Number of rigid-body modes
s	Reference semispan, ft (i.e., span measured from root to tip)
U	Element of dynamic matrix, in.
V	Velocity, knots
W	Element of aerodynamic weighting matrix, dimensionless

SYMBOLS (continued)

λ	Eigenvalue, $\lambda = \lambda_R + i\lambda_I$, in.
ρ	Atmospheric density, slugs/ft^3
f	Frequency, cps

Matrix Notation

$\lceil\ \rceil$ Square

$\{\ \}$ Column

$[\ ]^T$ Transposed

$\lceil I \rfloor$ Unit

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) HUGHES AIRCRAFT COMPANY, MISSILE SYSTEMS DIVISION FALLBROOK AND ROSCOE BLVDS. CANOGA PARK, CALIFORNIA 91304	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

COLLOCATION FLUTTER ANALYSIS STUDY

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

FINAL REPORT (MARCH 1968 THROUGH MARCH 1969)

5. AUTHOR(S) (First name, middle initial, last name)

DYNAMICS AND ENVIRONMENT SECTION, DONALD R. ULBRICH

6. REPORT DATE APRIL 4, 1969	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. N00019-68-C-0247 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

IN ADDITION TO SECURITY REQUIREMENTS WHICH APPLY TO THIS DOCUMENT AND MUST BE MET, EACH TRANSMITTAL OF THIS DOCUMENT OUTSIDE THE AGENCIES OF THE U.S. GOVERNMENT MUST HAVE PRIOR APPROVAL OF THE COMMANDER NASC

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY NAVAL AIR SYSTEMS COMMANDS DEPARTMENT OF THE NAVY WASHINGTON, D.C.

13. ABSTRACT

THIS STUDY COVERS THE DEVELOPMENT OF A SET OF COMPUTER PROGRAM TO PERFORM FLUTTER ANALYSIS BY THE COLLOCATION METHOD. WHILE THIS METHOD HAS BEEN KNOWN FOR SOME TIME, ONLY RECENTLY HAVE ADVANCES IN COMPUTER TECHNOLOGY MADE THE METHOD TECHNICALLY AND FINANCIALLY FEASIBLE. THE INGREDIENTS OF A COLLOCATION FLUTTER ANALYSIS ARE 1) A FLEXIBILITY MATRIX, 2) AERODYNAMIC INFLUENCE COEFFICIENT MATRIX, AND 3) AN EIGENVALUE SOLUTION. THIS STUDY IS PRESENTED IN FOUR VOLUMES. VOLUME I CONTAINS A GENERAL PROGRAM DISCUSSION. VOLUME II CONTAINS THE PROGRAM FLUENC WHICH CALCULATES THE FLEXIBILITY MATRIX. VOLUME III CONTAINS A SET OF THREE PROGRAMS TO CALCULATE AERODYNAMIC INFLUENCE COEFFICIENTS FOR SUBSONIC, TRANSONIC, AND SUPERSONIC FLIGHT REGIMES. VOLUME IV CONTAINS THE PROGRAM COFA WHICH SETS UP AND SOLVES THE FLUTTER EIGENVALUE MATRIX.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
FLUTTER						
VIBRATION						
AERODYNAMIC INFLUENCE COEFFICIENTS						